**Workflow Management**

**Software Test Report**

*Submitted in partial fulfillment of the requirements of the course CS223 – Software Engineering*

*Submitted by*

Chitraksh Sadayat (B16CS007)

Vishakh Suresh (B16CS038)

**VERSION HISTORY**

|  |  |  |
| --- | --- | --- |
| **Version #** | **Revision Date** | **Brief Description** |
| 1.0 | 29/03/2018 | Unit testing and Integration Testing |
|  |  |  |
|  |  |  |

**TABLE OF CONTENTS**

1. INTRODUCTION …………………………………………………………………… 4

* Purpose ………………………………………………………………………….. 4

1. TEST PLAN ………………………………………………………………………..… 4
2. DETAILS OF UNIT TESTING ……………………………………………………… 5
3. DETAILS OF SYSTEM TESTING ………………………………………………….. 57
4. APPENDIX (Control Flow Graphs) ………………………………………………….. 65
5. APPENDIX (Glimpses into the execution) …………………………………………… 78

# 1 INTRODUCTION

## 1.1 Purpose

This *Workflow Management Unit and System* Test Report provides a summary of the results of test performed as outlined within this document. Testing is done to eliminate faults and errors to avoid user-end failure of the software.

Tests are performed under following specifications:

1. **In-scope**:

The functionalities of following modules are tested:

1. Apply leave: for Undergrads, research scholars, faculty, HOD, director
2. Apply finance: for Undergrads, research scholar
3. Approve leave: for faculty, HOD, director, administrator
4. Approve finance: for faculty, administrator
5. Check leave status: for Undergrads, research scholars, faculty, HOD, director
6. Check finance status: for Undergrads, research scholar
7. Register
8. Login
9. **Out-of-scope**:

Performance related tests including efficiency and space-time complexities has not been performed.

1. **Items not tested**:

The storing and retrieval of the objects during starting and shutting down of system is not done since file handling has not been implemented yet.

# 2 TEST PLAN

1. For each independent module unit testing is performed.
2. For modules from ‘a’ to ‘f’ Control flow testing is applied which includes complete branch and predicate coverage.
3. The modules involve procedural-call interface and shared-memory interface. So, attempts are made to perform appropriate system integration testing: both pairwise and end-to-end testing to minimize errors.

# 3 DETAILS OF UNIT TESTING

* 1. **APPLY LEAVE: FOR UNDERGRADS**

|  |  |
| --- | --- |
| **Function Name** | apply\_leave |
| **Class** | Undergrads |
| **Input Parameters** | **a (int)** – Number of leaves the user has applied for |
| **obj (Leave**) – Leave object with details of designation, number of pending leaves, leave application status, etc |
| **Processing** | * Checks if ‘a’ leaves can be granted or not * If yes, leave is applied and the application is forwarded to the Administrator for approval. * If not, an error message is given to user. |
| **Output Parameters** | None |
| **Code Snippet** | |
| C:\Users\DELL\Pictures\Screenshots\Screenshot (98).png | |

**3.1.1 Test items**

The unit to be tested here is the function apply\_leave (which facilitates the process of applying a leave), for an undergraduate student.

**3.1.2 Features to be tested**

* Validity of the number of leaves applied .i.e. input parameter
* Validity of the leave application .i.e. whether the requested number of leaves can be granted or not.

**3.1.3 Item pass/ fail criteria**

* If ‘a’ is less than zero: the leave request is rejected by the system and a message is sent back to the applicant.

*Rationale*: Number of leaves applied for cannot be negative.

* Once it is verified that the number of leaves applied for is positive, the system checks if ‘a’ leaves can be granted or not.
* If a > number of pending leaves, then the application is rejected with an appropriate error message to the applicant.
* If a <= number of pending leaves, then the application is accepted and forwarded to the administrator for approval.
* If the system abides by these criteria, the test is deemed to have been passed.

**3.1.4 Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Input** | **Local Variable values** | **Expected Output** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| **a** | **b** |
| 3.1.4.1 | 12 | 50 | Leave applied | Leave applied | **PASS** | - |
| 3.1.4.2 | 40 | 38 | Not applied | Not applied | **PASS** | - |
| 3.1.4.3 | 20 | 38 | Leave applied | Leave applied | **PASS** | - |
| 3.1.4.4 | 100 | 50 | Not applied | Not applied | **PASS** | - |
| 3.1.4.5 | -20 | 50 | Not applied | Not applied | **PASS** | - |
| 3.1.4.6 | -20 | 38 | Not applied | Not applied | **PASS** | - |

**3.1.5 Test Result**

The function apply\_leave for Undergrads processed the inputs as expected for all the input cases.

**3.2 APPLY LEAVE: FOR RESEARCH SCHOLAR**

|  |  |
| --- | --- |
| **Function Name** | apply\_leave |
| **Class** | Research\_scholar |
| **Input Parameters** | **a (int)** – Number of leaves the user has applied for |
| **obj (Leave)** – Leave object with details of designation, number of pending leaves, leave application status, etc |
| **Processing** | * Checks if ‘a’ leaves can be granted or not * If yes, leave is applied and the application is forwarded to the faculty (Research supervisor) for approval. * If not, an error message is given to user. |
| **Output Parameters** | None |
| **Code Snippet** | |
| C:\Users\DELL\Pictures\Screenshots\Screenshot (103).png | |

**3.2.1 Test items**

The unit to be tested here is the function apply\_leave (which facilitates the process of applying a leave), for a research scholar.

**3.2.2 Features to be tested**

* Validity of the number of leaves applied .i.e. input parameter
* Validity of the leave application .i.e. whether the requested number of leaves can be granted or not.

**3.2.3 Item pass/ fail criteria**

* If ‘a’ is less than zero: the leave request is rejected by the system and a message is sent back to the applicant.

*Rationale*: Number of leaves applied for cannot be negative.

* Once it is verified that the number of leaves applied for is positive, the system checks if ‘a’ leaves can be granted or not.
* If a > number of pending leaves, then the application is rejected with an appropriate error message to the applicant.
* If a <= number of pending leaves, then the application is accepted and forwarded to the faculty (Research supervisor) for approval.
* If the system abides by these criteria, the test is deemed to have been passed.

**3.2.4 Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Input** | **Local Variable values** | **Expected Output** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| **a** | **b** |
| 3.2.4.1 | 12 | 50 | Leave applied | Leave applied | **PASS** | - |
| 3.2.4.2 | 40 | 38 | Not applied | Not applied | **PASS** | - |
| 3.2.4.3 | 20 | 38 | Leave applied | Leave applied | **PASS** | - |
| 3.2.4.4 | 100 | 50 | Not applied | Not applied | **PASS** | - |
| 3.2.4.5 | -20 | 50 | Not applied | Not applied | **PASS** | - |
| 3.2.4.6 | -20 | 38 | Not applied | Not applied | **PASS** | - |

**3.2.5 Test Result**

The function apply\_leave for Research\_scholar processed the inputs as expected for all the input cases.

|  |  |
| --- | --- |
| **Function Name** | apply\_leave |
| **Class** | Faculty |
| **Input Parameters** | **a (int)** – Number of leaves the user has applied for |
| **obj (Leave)** – Leave object with details of designation, number of pending leaves, leave application status, etc |
| **Processing** | * Checks if ‘a’ leaves can be granted or not * If yes, leave is applied. * If the object obj is a Research scholar object, the application is forwarded to the Administrator for approval; else it is forwarded to the HOD of the concerned department for his approval. * If not, an error message is given to user. |
| **Output Parameters** | None |
| **Code Snippet** | |
| C:\Users\DELL\Pictures\Screenshots\Screenshot (110).png | |

**3.3 APPLY LEAVE: FOR FACULTY**

**3.3.1 Test items**

The unit to be tested here is the function apply\_leave (which facilitates the process of applying a leave), for a faculty.

**3.3.2 Features to be tested**

* Validity of the number of leaves applied .i.e. input parameter
* Validity of the leave application .i.e. whether the requested number of leaves can be granted or not.

**3.3.3 Item pass/ fail criteria**

* If ‘a’ is less than zero: the leave request is rejected by the system and a message is sent back to the applicant.

*Rationale*: Number of leaves applied for cannot be negative.

* Once it is verified that the number of leaves applied for is positive, the system checks if ‘a’ leaves can be granted or not.
* If a > number of pending leaves, then the application is rejected with an appropriate error message to the applicant.
* If a <= number of pending leaves and
* obj is a Faculty object, then the application is accepted and forwarded to the HOD of the concerned department for approval.
* obj is a Research\_scholar object, then the application is accepted and forwarded to the administrator for approval. (This case occurs when a Faculty tries to approve the leave application of a research scholar under him.)
* For the cases when ‘a’ is negative and ‘a’ exceeds the number of pending leaves, explicit testing for the situation when obj is a Research\_scholar object; is NOT required.

*Rationale*: These 2 situations would already have been tested in the apply\_leave function for Research\_scholar when they try to apply for one.

* If the system abides by these criteria, the test is deemed to have been passed.

**3.3.4 Test Cases**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Input** | | **Local Variable values** | **Expected Output** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| **a** | **obj** | **b** |
| 3.3.4.1 | 12 | Fac\* | 50 | Leave applied  (forwarded to HOD) | Leave applied  (forwarded to HOD) | **PASS** | - |
| 3.3.4.2 | 40 | Fac\* | 38 | Not applied | Not applied | **PASS** | - |
| 3.3.4.3 | 20 | Fac\* | 38 | Leave applied  (forwarded to HOD) | Leave applied  (forwarded to HOD) | **PASS** | - |
| 3.3.4.4 | 12 | RS# | 50 | Leave applied  (forwarded to Administrator) | Leave applied  (forwarded to Administrator) | **PASS** | - |
| 3.3.4.5 | 20 | RS# | 38 | Leave applied  (forwarded to Administrator) | Leave applied  (forwarded to Administrator) | **PASS** | - |
| 3.3.4.6 | 100 | Fac\* | 50 | Not applied | Not applied | **PASS** | - |
| 3.3.4.7 | -20 | Fac\* | 50 | Not applied | Not applied | **PASS** | - |
| 3.3.4.8 | -20 | Fac\* | 38 | Not applied | Not applied | **PASS** | - |

(\*) Fac : Faculty

(#) RS : Research Scholar

**3.3.5 Test Result**

The function apply\_leave for Faculty processed the inputs as expected for all the input cases.

**3.4 APPLY LEAVE: FOR HOD**

|  |  |
| --- | --- |
| **Function Name** | apply\_leave |
| **Class** | HOD |
| **Input Parameters** | **a (int)** – Number of leaves the user has applied for |
| **obj (Leave)** – Leave object with details of designation, number of pending leaves, leave application status, etc |
| **Processing** | * Checks if ‘a’ leaves can be granted or not * If yes, leave is applied. * If the object obj is a Faculty object, the application is forwarded to the Director for his approval. * If obj is a HOD object it is again forwarded to the Director for his approval. * If not, an error message is given to user. |
| **Output Parameters** | None |
| **Code Snippet** | |
| C:\Users\DELL\Pictures\Screenshots\Screenshot (112).png | |

**3.4.1 Test items**

The unit to be tested here is the function apply\_leave (which facilitates the process of applying a leave), for a HOD.

**3.4.2 Features to be tested**

* Validity of the number of leaves applied .i.e. input parameter
* Validity of the leave application .i.e. whether the requested number of leaves can be granted or not.

**3.4.3 Item pass/ fail criteria**

* If ‘a’ is less than zero: the leave request is rejected by the system and a message is sent back to the applicant.

*Rationale*: Number of leaves applied for cannot be negative.

* Once it is verified that the number of leaves applied for is positive, the system checks if ‘a’ leaves can be granted or not.
* If a > number of pending leaves, then the application is rejected with an appropriate error message to the applicant.
* If a <= number of pending leaves and
* obj is a HOD object, then the application is accepted and forwarded to the Director for approval.
* obj is a Faculty object, then the application is accepted and forwarded again to the Director for approval. (This case occurs when a HOD tries to approve the leave application of a faculty in his department.)
* For the cases when ‘a’ is negative and ‘a’ exceeds the number of pending leaves, explicit testing for the situation when obj is a Faculty object; is NOT required.

*Rationale*: These 2 situations would already have been tested in the apply\_leave function for Faculty when they try to apply for one.

* If the system abides by these criteria, the test is deemed to have been passed.

**3.4.4 Test Cases**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Input** | | **Local Variable values** | **Expected Output** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| **a** | **obj** | **b** |
| 3.4.4.1 | 12 | HOD | 50 | Leave applied  (forwarded to Director) | Leave applied  (forwarded to Director ) | **PASS** | - |
| 3.4.4.2 | 40 | HOD | 38 | Not applied | Not applied | **PASS** | - |
| 3.4.4.3 | 20 | HOD | 38 | Leave applied  (forwarded to Director ) | Leave applied  (forwarded to Director ) | **PASS** | - |
| 3.4.4.4 | 12 | Fac\* | 50 | Leave applied  (forwarded to Director ) | Leave applied  (forwarded to Director ) | **PASS** | - |
| 3.4.4.5 | 20 | Fac\* | 38 | Leave applied  (forwarded to Director ) | Leave applied  (forwarded to Director ) | **PASS** | - |
| 3.4.4.6 | 100 | HOD | 50 | Not applied | Not applied | **PASS** | - |
| 3.4.4.7 | -20 | HOD | 50 | Not applied | Not applied | **PASS** | - |
| 3.4.4.8 | -20 | HOD | 38 | Not applied | Not applied | **PASS** | - |

(\*) Fac : Faculty

**3.4.5 Test Result**

The function apply\_leave for HOD processed the inputs as expected for all the input cases.

**3.5 APPLY LEAVE: FOR DIRECTOR**

|  |  |
| --- | --- |
| **Function Name** | apply\_leave |
| **Class** | Director |
| **Input Parameters** | **a (int)** – Number of leaves the user has applied for |
| **obj (Leave)** – Leave object with details of designation, number of pending leaves, leave application status, etc |
| **Processing** | * Checks if ‘a’ leaves can be granted or not * If yes, leave is applied. * If the object obj is a Director object, the application is forwarded to the Administrator for approval * If obj is a HOD or a Faculty object it is again forwarded to the Administrator for approval. * If not, an error message is given to user. |
| **Output Parameters** | None |

**3.5.1 Test items**

The unit to be tested here is the function apply\_leave (which facilitates the process of applying a leave), for the Director.

**3.5.2 Features to be tested**

* Validity of the number of leaves applied .i.e. input parameter
* Validity of the leave application .i.e. whether the requested number of leaves can be granted or not.

**3.5.3 Item pass/ fail criteria**

* If ‘a’ is less than zero: the leave request is rejected by the system and a message is sent back to the applicant.

*Rationale*: Number of leaves applied for cannot be negative.

* Once it is verified that the number of leaves applied for is positive, the system checks if ‘a’ leaves can be granted or not.
* If a > number of pending leaves, then the application is rejected with an appropriate error message to the applicant.
* If a <= number of pending leaves and
* obj is a HOD object or a Faculty object, then the application is accepted and forwarded to the Administrator for approval. (This case occurs when the Director tries to approve the leave application of a HOD or Faculty.)
* obj is a Director object, then the application is accepted and forwarded again to the Administrator for approval.
* For the cases when ‘a’ is negative and ‘a’ exceeds the number of pending leaves, explicit testing for the situation when obj is a HOD/Faculty object; is NOT required.

*Rationale*: These 2 situations would already have been tested in the apply\_leave function for HOD/Faculty when they try to apply for one.

* If the system abides by these criteria, the test is deemed to have been passed.

**Code Snippet**



|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Input** | | **Local Variable values** | **Expected Output** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| **a** | **obj** | **b** |
| 3.5.4.1 | 12 | Director | 50 | Leave applied  (forwarded to Administrator) | Leave applied  (forwarded to Administrator ) | **PASS** | - |
| 3.5.4.2 | 40 | Director | 38 | Not applied | Not applied | **PASS** | - |
| 3.5.4.3 | 20 | Director | 38 | Leave applied  (forwarded to Administrator ) | Leave applied  (forwarded to Administrator ) | **PASS** | - |
| 3.5.4.4 | 12 | HOD | 50 | Leave applied  (forwarded to Administrator ) | Leave applied  (forwarded to Administrator ) | **PASS** | - |
| 3.5.4.5 | 20 | HOD | 38 | Leave applied  (forwarded to Administrator ) | Leave applied  (forwarded to Administrator ) | **PASS** | - |
| 3.5.4.6 | 12 | Fac\* | 50 | Leave applied  (forwarded to Administrator ) | Leave applied  (forwarded to Administrator ) | **PASS** | - |
| 3.5.4.7 | 20 | Fac\* | 38 | Leave applied  (forwarded to Administrator ) | Leave applied  (forwarded to Administrator ) | **PASS** | - |
| 3.5.4.8 | 100 | Director | 50 | Not applied | Not applied | **PASS** | - |
| 3.5.4.9 | -20 | Director | 50 | Not applied | Not applied | **PASS** | - |
| 3.5.4.10 | -20 | Director | 38 | Not applied | Not applied | **PASS** | - |

**3.5.4 Test Cases**

(\*) Fac : Faculty

**3.5.5 Test Result**

The function apply\_leave for the Director processed the inputs as expected for all the input cases.

|  |  |
| --- | --- |
| **Function Name** | apply\_finance |
| **Class** | Undergrads |
| **Input Parameters** | **a (double)** – The financial assistance the user has applied for |
| **obj (Finance)** – Finance object with details of financial income, amount received currently, financial application status, etc |
| **Processing** | * Checks if ‘a’ amount can be granted or not (amount ‘a’ can be granted if the total assistance does not exceed 2 lakhs) * If yes, financial assistance is applied and the application is forwarded to the Administrator for approval. * If not, an error message is given to user. |
| **Output Parameters** | None |
| **Code Snippet** | |
| C:\Users\DELL\Pictures\Screenshots\Screenshot (101).png | |

* 1. **APPLY FINANCE: FOR UNDERGRADS**

**3.6.1 Test items**

The unit to be tested here is the function apply\_finance (which facilitates the process of applying for financial assistance), for an undergraduate student.

**3.6.2 Features to be tested**

* Validity of the amount applied for .i.e. input parameter
* Validity of the application .i.e. whether the amount requested can be granted or not.

**3.6.3 Item pass/ fail criteria**

* If ‘a’ is less than zero: the request is rejected by the system and a message is sent back to the applicant.

*Rationale*: Amount applied for cannot be negative.

* Once it is verified that the amount applied for is positive, the system checks if the amount ‘a’ can be granted or not.
* If a + Amount received now as financial assistance > 2,00,000 (OR) family income exceeds 5 lakhs, then the application is rejected with an appropriate error message to the applicant.
* If a + Amount received now as financial assistance <= 2,00,000 (AND) family income is below 5 lakhs, then the application is accepted and forwarded to the Administrator for approval.
* If the system abides by these criteria, the test is deemed to have been passed.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Input** | **Local Variables** | **Family Income** | **Expected Output** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| **a** | **b** |
| 3.6.4.1 | 1,00,000 | 0 | 2,00,000 | Applied | Applied | **PASS** | - |
| 3.6.4.2 | 1,50,000 | 1,00,000 | 2,00,000 | Not applied | Not applied | **PASS** | - |
| 3.6.4.3 | 50,000 | 1,00,000 | 2,00,000 | Applied | Applied | **PASS** | - |
| 3.6.4.4 | 10,00,000 | 0 | 2,00,000 | Not applied | Not applied | **PASS** | - |
| 3.6.4.5 | -20,000 | 0 | 2,00,000 | Not applied | Not applied | **PASS** | - |
| 3.6.4.6 | -20,000 | 1,00,000 | 2,00,000 | Not applied | Not applied | **PASS** | - |
| 3.6.4.7 | 1,00,000 | 0 | 8,00,000 | Not applied | Not applied | **PASS** | - |
| 3.6.4.8 | 1,50,000 | 1,00,000 | 8,00,000 | Not applied | Not applied | **PASS** | - |
| 3.6.4.9 | 50,000 | 1,00,000 | 8,00,000 | Not applied | Not applied | **PASS** | - |
| 3.6.4.10 | 10,00,000 | 0 | 8,00,000 | Not applied | Not applied | **PASS** | - |
| 3.6.4.11 | -20,000 | 0 | 8,00,000 | Not applied | Not applied | **PASS** | - |
| 3.6.4.12 | -20,000 | 1,00,000 | 8,00,000 | Not applied | Not applied | **PASS** | - |

**3.6.4 Test Cases**

**3.6.5 Test Result**

The function apply\_finance for Undergrads processed the inputs as expected for all the input cases.

* 1. **APPLY FINANCE: FOR RESEARCH SCHOLARS**

|  |  |
| --- | --- |
| **Function Name** | apply\_finance |
| **Class** | Research\_scholar |
| **Input Parameters** | **a (double)** – The financial assistance the user has applied for |
| **obj (Finance)** –Finance object with details of financial income, amount received currently, financial application status, etc |
| **Processing** | * Checks if ‘a’ amount can be granted or not (amount ‘a’ can be granted if the total assistance does not exceed 2 lakhs) * If yes, financial assistance is applied and the application is forwarded to the faculty (Research supervisor) for approval. * If not, an error message is given to user. |
| **Output Parameters** | None |
| **Code Snippet** | |
| **C:\Users\DELL\Pictures\Screenshots\Screenshot (105).png** | |

**3.7.1 Test items**

The unit to be tested here is the function apply\_finance (which facilitates the process of applying for financial assistance), for a research scholar.

**3.7.2 Features to be tested**

* Validity of the amount applied for .i.e. input parameter
* Validity of the application .i.e. whether the amount requested can be granted or not.

**3.7.3 Item pass/ fail criteria**

* If ‘a’ is less than zero: the request is rejected by the system and a message is sent back to the applicant.

*Rationale*: Amount applied for cannot be negative.

* Once it is verified that the amount applied for is positive, the system checks if the amount ‘a’ can be granted or not.
* If a + Amount received now as financial assistance > 2,00,000 (OR) family income exceeds 5 lakhs, then the application is rejected with an appropriate error message to the applicant.
* If a + Amount received now as financial assistance <= 2,00,000 (AND) family income is below 5 lakhs, then the application is accepted and forwarded to the faculty (Research supervisor) for approval.
* If the system abides by these criteria, the test is deemed to have been passed.

**3.7.4 Test Cases**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Test Input** | **Local Variables** | **Family Income** | **Expected Output** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| **a** | **b** |
| 3.7.4.1 | 1,00,000 | 0 | 2,00,000 | Applied | Applied | **PASS** | - |
| 3.7.4.2 | 1,50,000 | 1,00,000 | 2,00,000 | Not applied | Not applied | **PASS** | - |
| 3.7.4.3 | 50,000 | 1,00,000 | 2,00,000 | Applied | Applied | **PASS** | - |
| 3.7.4.4 | 10,00,000 | 0 | 2,00,000 | Not applied | Not applied | **PASS** | - |
| 3.7.4.5 | -20,000 | 0 | 2,00,000 | Not applied | Not applied | **PASS** | - |
| 3.7.4.6 | -20,000 | 1,00,000 | 2,00,000 | Not applied | Not applied | **PASS** | - |
| 3.7.4.7 | 1,00,000 | 0 | 8,00,000 | Not applied | Not applied | **PASS** | - |
| 3.7.4.8 | 1,50,000 | 1,00,000 | 8,00,000 | Not applied | Not applied | **PASS** | - |
| 3.7.4.9 | 50,000 | 1,00,000 | 8,00,000 | Not applied | Not applied | **PASS** | - |
| 3.7.4.10 | 10,00,000 | 0 | 8,00,000 | Not applied | Not applied | **PASS** | - |
| 3.7.4.11 | -20,000 | 0 | 8,00,000 | Not applied | Not applied | **PASS** | - |
| 3.7.4.12 | -20,000 | 1,00,000 | 8,00,000 | Not applied | Not applied | **PASS** | - |

**3.7.5 Test Result**

The function apply\_finance for Research\_scholar processed the inputs as expected for all the input cases.

* 1. **APPROVE LEAVE: FOR FACULTY**

|  |  |
| --- | --- |
| **Function Name** | approve\_leave |
| **Class** | Faculty |
| **Input Parameters** | **obj (Leave)** – Leave object with details of designation, number of pending leaves, leave application status, etc |
| **Processing** | * Checks if the leave application corresponding to obj is pending for approval or not. * If the leave application is pending for approval, then the control is handed over to the apply\_leave function (Faculty) using the object obj as the parameter. (Once the control is in the apply\_leave function, the path corresponding to a Research\_scholar is executed.) * The leave application corresponding to object obj is removed from the list of applications pending for the faculty’s approval. |
| **Output Parameters** | None |
| **Code Snippet** | |
| **C:\Users\DELL\Pictures\Screenshots\Screenshot (120).png** | |

**3.8.1 Test items**

The unit to be tested here is the function approve\_leave (which facilitates the process of approving a leave), for a faculty.

**3.8.2 Features to be tested**

* The primary feature to be tested for this function is the presence or absence of outstanding leave applications

**3.8.3 Item pass/ fail criteria**

* If the leave application corresponding to obj is not pending for approval, an appropriate message is given to the user.
* If the leave application corresponding to object obj is pending for the Faculty’s approval, then:
* apply\_leave function (for Faculty) is invoked with obj (Leave object for Research scholar) as its parameter.
* The research scholar’s application is processed and forwarded to the Administrator for approval.
* The list of pending leave applications is updated by removing this application from it.
* If the system abides by these criteria, the test is deemed to have been passed.

**3.8.4 Test Cases**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **#** | **User input**  *(Explicitly provided)* | | **Input Parameter**  *(Implicitly computed)* | **Expected Output** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| **user\_id** | **designation** | **obj** |
| 3.8.4.1 | Nil | Nil | Nil | No pending applications | No pending applications | **PASS** | - |
| 3.8.4.2 | 5 | Research\_  scholar | Research\_  scholar | Leave approved and forwarded | Leave approved and forwarded | **PASS** | - |
| 3.8.4.3 | 5 | HOD (Wrong designation) | - | Not found | Not found | **PASS** | - |
| 3.8.4.4 | 7  (Wrong  ID) | Research\_  scholar | - | Not found | Not found | **PASS** | - |
| 3.8.4.5 | 7  (Wrong  ID) | HOD (Wrong designation) | - | Not found | Not found | **PASS** | - |
| 3.8.4.6 | 3 | Research\_  scholar | Research\_  scholar | Leave approved and forwarded | Leave approved and forwarded | **PASS** | - |
| 4 | Research\_  scholar | Research\_  scholar | Leave approved and forwarded | Leave approved and forwarded |

**#** - The list of outstanding applications used for unit testing;

described in the table that follows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **# - Pending Leave Applications** | | | | |
| **Test ID** | **User ID** | **Applied number of leaves** | **Designation** | **Comments** |
| 3.8.4.1 | Nil | Nil | Nil | No Pending Applications |
| 3.8.4.2 | 5 | 12 | Research\_scholar | 1 pending application |
| 3.8.4.3 | 5 | 12 | Research\_scholar | 1 pending application |
| 3.8.4.4 | 5 | 12 | Research\_scholar | 1 pending application |
| 3.8.4.5 | 5 | 12 | Research\_scholar | 1 pending application |
| 3.8.4.6 | 3 | 15 | Research\_scholar | 2 pending applications |
| 4 | 18 | Research\_scholar |

**3.8.5** **Test Result**

The function approve\_leave for Faculty processed the inputs as expected for all the input cases.

* 1. **APPROVE LEAVE: FOR HOD**

|  |  |
| --- | --- |
| **Function Name** | approve\_leave |
| **Class** | HOD |
| **Input Parameters** | **obj (Leave)** – Leave object with details of designation, number of pending leaves, leave application status, etc |
| **Processing** | * Checks if the leave application corresponding to obj is pending for approval or not. * If the leave application is pending for approval, then the control is handed over to the apply\_leave function (HOD) using the object obj as the parameter. (Once the control is in the apply\_leave function, the path corresponding to a Faculty is executed.) * The leave application corresponding to object obj is removed from the list of applications pending for the HOD’s approval. |
| **Output Parameters** | None |
| **Code Snippet** | |
| **C:\Users\DELL\Pictures\Screenshots\Screenshot (122).png** | |

**3.9.1 Test items**

The unit to be tested here is the function approve\_leave (which facilitates the process of approving a leave), for a department HOD.

**3.9.2 Features to be tested**

* The primary feature to be tested for this function is the presence or absence of outstanding leave applications

**3.9.3 Item pass/ fail criteria**

* If the leave application corresponding to obj is not pending for approval, an appropriate message is given to the user.
* If the leave application corresponding to object obj is pending for the HOD’s approval, then:
* apply\_leave function (for HOD) is invoked with obj (leave object for Faculty) as its parameter.
* The Faculty’s application is processed and forwarded to the Director for approval.
* The list of pending leave applications is updated by removing this application from it.
* If the system abides by these criteria, the test is deemed to have been passed.

**3.9.4 Test Cases**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **#** | **User input**  *(Explicitly provided)* | | **Input Parameter**  *(Implicitly computed)* | **Expected Output** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| **user\_id** | **designation** | **obj** |
| 3.9.4.1 | Nil | Nil | Nil | No pending applications | No pending applications | **PASS** | - |
| 3.9.4.2 | 5 | Faculty | Faculty | Leave approved and forwarded | Leave approved and forwarded | **PASS** | - |
| 3.9.4.3 | 5 | Director (Wrong designation) | - | Not found | Not found | **PASS** | - |
| 3.9.4.4 | 7  (Wrong  ID) | Faculty | - | Not found | Not found | **PASS** | - |
| 3.9.4.5 | 7  (Wrong  ID) | Director (Wrong designation) | - | Not found | Not found | **PASS** | - |
| 3.9.4.6 | 3 | Faculty | Faculty | Leave approved and forwarded | Leave approved and forwarded | **PASS** | - |
| 4 | Faculty | Faculty | Leave approved and forwarded | Leave approved and forwarded |

**#** - The list of outstanding applications used for unit testing;

described in the table that follows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **# - Pending Leave Applications** | | | | |
| **Test ID** | **User ID** | **Applied number of leaves** | **Designation** | **Comments** |
| 3.9.4.1 | Nil | Nil | Nil | No Pending Applications |
| 3.9.4.2 | 5 | 12 | Faculty | 1 pending application |
| 3.9.4.3 | 5 | 12 | Faculty | 1 pending application |
| 3.9.4.4 | 5 | 12 | Faculty | 1 pending application |
| 3.9.4.5 | 5 | 12 | Faculty | 1 pending application |
| 3.9.4.6 | 3 | 15 | Faculty | 2 pending applications |
| 4 | 18 | Faculty |

**3.9.5** **Test Result**

The function approve\_leave for HOD processed the inputs as expected for all the input cases.

* 1. **APPROVE LEAVE: FOR DIRECTOR**

|  |  |
| --- | --- |
| **Function Name** | approve\_leave |
| **Class** | Director |
| **Input Parameters** | **obj (Leave)** – Leave object with details of designation, number of pending leaves, leave application status, etc |
| **Processing** | * Checks if the leave application corresponding to obj is pending for approval or not. * If the leave application is pending for approval, then the control is handed over to the apply\_leave function (HOD or Faculty) using the object obj as the parameter. (Once the control is in the apply\_leave function, the path corresponding to the HOD or Faculty is executed.) * The leave application corresponding to object obj is removed from the list of applications pending for the Director’s approval. |
| **Output Parameters** | None |
| **Code Snippet** | |
| **C:\Users\DELL\Pictures\Screenshots\Screenshot (124).png** | |

**3.10.1 Test items**

The unit to be tested here is the function approve\_leave (which facilitates the process of approving a leave), for the Director.

**3.10.2 Features to be tested**

* The primary feature to be tested for this function is the presence or absence of outstanding leave applications

**3.10.3 Item pass/ fail criteria**

* If the leave application corresponding to obj is not pending for approval, an appropriate message is given to the user.
* If the leave application corresponding to object obj is pending for the Director’s approval, then:
* apply\_leave function (for the Director) is invoked with obj (leave object for HOD or Faculty) as its parameter.
* The HOD’s or Faculty’s application is processed and forwarded to the Administrator for approval.
* The list of pending leave applications is updated by removing this application from it.
* If the system abides by these criteria, the test is deemed to have been passed.

**3.10.4 Test Cases**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **#** | **User input**  *(Explicitly provided)* | | **Input Parameter**  *(Implicitly computed)* | **Expected Output** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| **user\_id** | **designation** | **obj** |
| 3.10.4.1 | Nil | Nil | Nil | No pending applications | No pending applications | **PASS** | - |
| 3.10.4.2 | 5 | Faculty | Faculty | Leave approved and forwarded | Leave approved and forwarded | **PASS** | - |
| 3.10.4.3 | 5 | Director (Wrong designation) | - | Not found | Not found | **PASS** | - |
| 3.10.4.4 | 7  (Wrong  ID) | Faculty | - | Not found | Not found | **PASS** | - |
| 3.10.4.5 | 7  (Wrong  ID) | Director (Wrong designation) | - | Not found | Not found | **PASS** | - |
| 3.10.4.6 | 3 | Faculty | Faculty | Leave approved and forwarded | Leave approved and forwarded | **PASS** | - |
| 4 | Faculty | Faculty | Leave approved and forwarded | Leave approved and forwarded |
| 3.10.4.7 | 1 | HOD | HOD | Leave approved and forwarded | Leave approved and forwarded | **PASS** | - |
| 3.10.4.8 | 1 | Director (Wrong designation) | - | Not found | Not found | **PASS** | - |
| 3.10.4.9 | 9  (Wrong  ID) | HOD | - | Not found | Not found | **PASS** | - |
| 3.10.4.10 | 7  (Wrong  ID) | Director (Wrong designation) | - | Not found | Not found | **PASS** | - |
| 3.10.4.11 | 1 | HOD | HOD | Leave approved and forwarded | Leave approved and forwarded | **PASS** | - |
| 2 | HOD | HOD | Leave approved and forwarded | Leave approved and forwarded |
| 3.10.4.12 | 3 | Faculty | Faculty | Leave approved and forwarded | Leave approved and forwarded | **PASS** | - |
| 1 | HOD | HOD | Leave approved and forwarded | Leave approved and forwarded | **PASS** | - |

**#** - The list of outstanding applications used for unit testing;

described in the table that follows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **# - Pending Leave Applications** | | | | |
| **Test ID** | **User ID** | **Applied number of leaves** | **Designation** | **Comments** |
| 3.10.4.1 | Nil | Nil | Nil | No Pending Applications |
| 3.10.4.2 | 5 | 12 | Faculty@ | 1 pending application |
| 3.10.4.3 | 5 | 12 | Faculty@ | 1 pending application |
| 3.10.4.4 | 5 | 12 | Faculty@ | 1 pending application |
| 3.10.4.5 | 5 | 12 | Faculty@ | 1 pending application |
| 3.10.4.6 | 3 | 15 | Faculty@ | 2 pending applications |
| 4 | 18 | Faculty@ |
| 3.10.4.7 | 1 | 20 | HOD | 1 pending application |
| 3.10.4.8 | 1 | 20 | HOD | 1 pending application |
| 3.10.4.9 | 1 | 20 | HOD | 1 pending application |
| 3.10.4.10 | 1 | 20 | HOD | 1 pending application |
| 3.10.4.11 | 1 | 20 | HOD | 2 pending applications |
| 2 | 25 | HOD |
| 3.10.4.12 | 3 | 15 | Faculty@ | 2 pending applications |
| 1 | 20 | HOD |

@ - After approval by the corresponding department HOD

**3.10.5** **Test Result**

The function approve\_leave for the Director processed the inputs as expected for all the input cases.

|  |  |
| --- | --- |
| **Function Name** | approve\_leave |
| **Class** | Administrator |
| **Input Parameters** | **obj (Leave)** – Leave object with details of designation, number of pending leaves, leave application status, etc |
| **Processing** | * Checks if the leave application corresponding to obj is pending for approval or not. * If the leave application corresponding to obj (Undergrad/ Research scholar/ Faculty/ HOD/ Director) is pending for approval, then the application is approved. * The leave application corresponding to object obj is removed from the list of applications pending for the Administrator’s approval. * Set the field corresponding to the number of applied leaves, in obj to zero. * Corresponding changes are made in the database as well. |
| **Output Parameters** | None |

**3.11 APPROVE LEAVE: FOR ADMINISTRATOR**

**3.11.1 Test items**

The unit to be tested here is the function approve\_leave (which facilitates the process of approving a leave), for the Administrator.

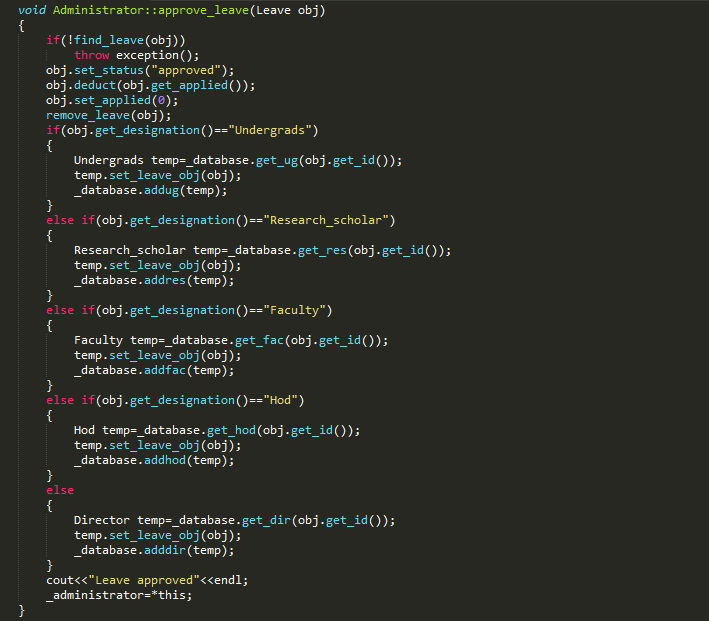
**3.11.2 Features to be tested**

* The primary feature to be tested for this function is the presence or absence of outstanding leave applications

**3.11.3 Item pass/ fail criteria**

* If the leave application corresponding to obj is not pending for approval, an appropriate message is given to the user.
* If the leave application corresponding to object obj is pending for the Administrator’s approval, then:
* The application is approved.
* The applied leaves (awaiting approval) field for the object is set to zero.
* The list of pending leave applications is updated by removing this application from it.
* If the system abides by these criteria, the test is deemed to have been passed.

**Code Snippet:**

****

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **#** | **User input**  *(Explicitly provided)* | | **Input Parameter**  *(Implicitly computed)* | **Expected Output** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| **user\_id** | **designation** | **obj** |
| 3.11.4.1 | Nil | Nil | Nil | No applications pending for approval | No applications pending for approval | **PASS** | - |
| 3.11.4.2 | 123 | Director | Director | Leave approved | Leave approved | **PASS** | - |
| 3.11.4.3 | 123 | HOD (Wrong designation) | - | Not found | Not found | **PASS** | - |
| 3.11.4.4 | 1 | HOD | HOD | Leave approved | Leave approved | **PASS** | - |
| 3.11.4.5 | 7  (Wrong  ID) | HOD | - | Not found | Not found | **PASS** | - |
| 3.11.4.6 | 2 | Faculty | Faculty | Leave approved | Leave approved | **PASS** | - |
| 3.11.4.7 | 2 | HOD  (Wrong designation) | - | Not found | Not found | **PASS** | - |
| 3.11.4.8 | 3 | Research\_  scholar | Research\_  scholar | Leave approved | Leave approved | **PASS** | - |
| 3.11.4.9 | 7  (Wrong  ID) | Research\_  scholar | - | Not found | Not found | **PASS** | - |
| 3.11.4.10 | 4 | Undergrads | Undergrads | Leave approved | Leave approved | **PASS** | - |
| 3.11.4.11 | 7  (Wrong  ID) | Undergrads | - | Not found | Not found | **PASS** | - |
| 3.11.4.12 | 123 | Director | Director | Leave approved | Leave approved | **PASS** | - |
| 1 | HOD | HOD | Leave approved | Leave approved |
| 2 | Faculty | Faculty | Leave approved | Leave approved |
| 3 | Research\_  scholar | Research\_  scholar | Leave approved | Leave approved |
| 4 | Undergrads | Undergrads | Leave approved | Leave approved |

**3.11.4 Test Cases**

**#** - The list of outstanding applications used for unit testing; described in the table that follows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **# - Pending Leave Applications** | | | | |
| **Test ID** | **User ID** | **Applied number of leaves** | **Designation** | **Comments** |
| 3.11.4.1 | Nil | Nil | Nil | No Pending Applications |
| 3.11.4.2 | 123 | 25 | Director | 1 pending application |
| 3.11.4.3 | 123 | 25 | Director | 1 pending application |
| 3.11.4.4 | 1 | 21 | HOD a | 1 pending application |
| 3.11.4.5 | 1 | 21 | HOD a | 1 pending application |
| 3.11.4.6 | 2 | 18 | Faculty b | 1 pending application |
| 3.11.4.7 | 2 | 18 | Faculty b | 1 pending application |
| 3.11.4.8 | 3 | 15 | Research\_scholar c | 1 pending application |
| 3.11.4.9 | 3 | 15 | Research\_scholar c | 1 pending application |
| 3.11.4.10 | 4 | 12 | Undergrad d | 1 pending application |
| 3.11.4.11 | 4 | 12 | Undergrad d | 1 pending application |
| 3.11.4.12 | 123 | 25 | Director | 5 pending applications |
| 1 | 21 | HOD |
| 2 | 18 | Faculty |
| 3 | 15 | Research\_scholar |
| 4 | 12 | Undergrad |

**Legend:**

* HOD a, Faculty b, Research\_scholar c, Undergrad d – Applications that have reached the Administrator after approval by the concerned authorities in the defined hierarchy chain
* There are several other possibilities for the list of pending leave applications. Further for each of these cases, there are 2 possible situations depending on the user inputs:
* entered user\_id and designation are correct and matching.
* any one or both of the user\_id and designation are incorrect.

The application is approved if and only if both the inputs are correct.

Here 12 of those cases are shown to depict a few.

**3.11.5** **Test Result**

The function approve\_leave for the Administrator processed the inputs as expected for the selected input cases.

**3.12 APPROVE FINANCE: FOR FACULTY**

|  |  |
| --- | --- |
| **Function Name** | approve\_finance |
| **Class** | Faculty |
| **Input Parameters** | **obj (Finance)** – Finance object with details of financial income, amount received currently, financial application status, etc |
| **Processing** | * Checks if the application corresponding to obj is pending for approval or not. * If the finance application from a research scholar is pending for approval, the application is approved and forwarded to the Administrator for approval. * The leave application corresponding to object obj is removed from the list of applications pending for the faculty’s approval. |
| **Output Parameters** | None |
| **Code Snippet** | |
| **C:\Users\DELL\Pictures\Screenshots\Screenshot (130).png** | |

**3.12.1 Test items**

The unit to be tested here is the function approve\_finance (which facilitates the process of approving a financial application), for a faculty.

**3.12.2 Features to be tested**

* The primary feature to be tested for this function is the presence or absence of outstanding financial applications.

**3.12.3 Item pass/ fail criteria**

* If the application corresponding to obj is not pending for approval, an appropriate message is given to the user.
* If the leave application corresponding to object obj is pending for the Faculty’s approval, then:
* The research scholar’s application is processed and forwarded to the Administrator for approval.
* The list of pending financial applications is updated by removing this application from it.
* If the system abides by these criteria, the test is deemed to have been passed.

**3.12.4 Test Cases**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **#** | **User input**  *(Explicitly provided)* | | **Input Parameter**  *(Implicitly computed)* | **Expected Output** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| **user\_id** | **designation** | **obj** |
| 3.12.4.1 | Nil | Nil | Nil | No pending applications | No pending applications | **PASS** | - |
| 3.12.4.2 | 5 | Research\_  scholar | Research\_  scholar | Approved and forwarded | Approved and forwarded | **PASS** | - |
| 3.12.4.3 | 5 | HOD (Wrong designation) | - | Not found | Not found | **PASS** | - |
| 3.12.4.4 | 7  (Wrong  ID) | Research\_  scholar | - | Not found | Not found | **PASS** | - |
| 3.12.4.5 | 7  (Wrong  ID) | HOD (Wrong designation) | - | Not found | Not found | **PASS** | - |
| 3.12.4.6 | 3 | Research\_  scholar | Research\_  scholar | Approved and forwarded | Approved and forwarded | **PASS** | - |
| 4 | Research\_  scholar | Research\_  scholar | Approved and forwarded | Approved and forwarded |

**#** - The list of outstanding applications used for unit testing;

described in the table that follows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **# - Pending Financial Assistance Applications** | | | | |
| **Test ID** | **User ID** | **Applied amount** | **Designation** | **Comments** |
| 3.12.4.1 | Nil | Nil | Nil | No Pending Applications |
| 3.12.4.2 | 5 | 1,00,000 | Research\_scholar | 1 pending application |
| 3.12.4.3 | 5 | 1,00,000 | Research\_scholar | 1 pending application |
| 3.12.4.4 | 5 | 1,00,000 | Research\_scholar | 1 pending application |
| 3.12.4.5 | 5 | 1,00,000 | Research\_scholar | 1 pending application |
| 3.12.4.6 | 3 | 1,50,000 | Research\_scholar | 2 pending applications |
| 4 | 1,80,000 | Research\_scholar |

**3.12.5** **Test Result**

The function approve\_finance for Faculty processed the inputs as expected for all the input cases.

**3.13 APPROVE FINANCE: FOR ADMINISTRATOR**

|  |  |
| --- | --- |
| **Function Name** | approve\_leave |
| **Class** | Administrator |
| **Input Parameters** | **obj (Finance)** – Finance object with details of financial income, amount received currently, financial application status, etc |
| **Processing** | * Checks if the application corresponding to obj is pending for approval or not. * If the finance application from an undergrad or research scholar is pending for approval, the application is approved and the corresponding changes are made. * The leave application corresponding to object obj is removed from the list of applications pending for the Administrator’s approval. |
| **Output Parameters** | None |
| **Code Snippet** | |
| **C:\Users\DELL\Pictures\Screenshots\Screenshot (132).png** | |

**3.13.1 Test items**

The unit to be tested here is the function approve\_finance (which facilitates the process of approving a financial application), for the Administrator.

**3.13.2 Features to be tested**

* The primary feature to be tested for this function is the presence or absence of outstanding financial applications.

**3.13.3 Item pass/ fail criteria**

* If the application corresponding to obj is not pending for approval, an appropriate message is given to the user.
* If the leave application corresponding to object obj is pending for the Administrator’s approval, then:
* The Undergrad’s or research scholar’s application is approved.
* The list of pending financial applications is updated by removing this application from it.
* If the system abides by these criteria, the test is deemed to have been passed.

**3.13.4 Test Cases**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **#** | **User input**  *(Explicitly provided)* | | **Input Parameter**  *(Implicitly computed)* | **Expected Output** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| **user\_id** | **designation** | **obj** |
| 3.13.4.1 | Nil | Nil | Nil | No pending applications | No pending applications | **PASS** | - |
| 3.13.4.2 | 5 | Research\_  scholar | Research\_  scholar | Approved | Approved | **PASS** | - |
| 3.13.4.3 | 5 | HOD (Wrong designation) | - | Not found | Not found | **PASS** | - |
| 3.13.4.4 | 4 | Undergrads | Undergrads | Approved | Approved | **PASS** | - |
| 3.13.4.5 | 7  (Wrong  ID) | Undergrads | - | Not found | Not found | **PASS** | - |
| 3.13.4.6 | 1 | Undergrads | Undergrads | Approved | Approved | **PASS** | - |
| 2 | Research\_  scholar | Research\_  scholar | Approved | Approved |

**#** - The list of outstanding applications used for unit testing; described in the table that follows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **# - Pending Financial Assistance Applications** | | | | |
| **Test ID** | **User ID** | **Applied amount** | **Designation** | **Comments** |
| 3.13.4.1 | Nil | Nil | Nil | No Pending Applications |
| 3.13.4.2 | 5 | 1,00,000 | Research\_scholar | 1 pending application |
| 3.13.4.3 | 5 | 1,00,000 | Research\_scholar | 1 pending application |
| 3.13.4.4 | 4 | 1,50,000 | Undergrads | 1 pending application |
| 3.13.4.5 | 4 | 1,50,000 | Undergrads | 1 pending application |
| 3.13.4.6 | 1 | 1,25,000 | Undergrads | 2 pending applications |
| 2 | 1,80,000 | Research\_scholar |

**3.13.5** **Test Result**

The function approve\_finance for Administrator processed the inputs as expected for all the input cases.

.

**3.14 CHECK LEAVE STATUS**

**3.14.1 Test items**

The unit to be tested here is the function get\_status (which facilitates the process of getting a leave application status), for all the non-administrative users.

This function is placed in the class Leave and hence can be used to display the status for any user using the system.

* + 1. **Features to be tested**

The only feature to be tested for this function is the status it returns

**3.14.3 Item pass/ fail criteria**

If the correct status is displayed when invoked for a particular object, the test is deemed to have been passed.

|  |  |
| --- | --- |
| **Function Name** | get\_status |
| **Class** | Leave |
| **Invoked for** | **obj (Leave)** – Leave object with details of designation, number of pending leaves, leave application status, etc |
| **Input Parameters** | None |
| **Processing** | None (since it is defined to be just a getter function) |
| **Output Parameters** | Return the string stored in the status field of the Leave object |
| **Code Snippet** | |
| **C:\Users\DELL\Pictures\Screenshots\Screenshot (136).png** | |

**3.14.4 Test cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Invoked for** | **Actual Application**  **Status** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| 3.14.4.1 | Undergrads | Not applied | Not applied | **PASS** | - |
| 3.14.4.2 | Undergrads | Administrator | Administrator | **PASS** | - |
| 3.14.4.3 | Undergrads | Approved | Approved | **PASS** | - |
| 3.14.4.4 | Research\_scholar | Not applied | Not applied | **PASS** | - |
| 3.14.4.5 | Research\_scholar | Faculty | Faculty | **PASS** | - |
| 3.14.4.6 | Research\_scholar | Administrator | Administrator | **PASS** | - |
| 3.14.4.7 | Research\_scholar | Approved | Approved | **PASS** | - |
| 3.14.4.8 | Faculty | Not applied | Not applied | **PASS** | - |
| 3.14.4.9 | Faculty | HOD | HOD | **PASS** | - |
| 3.14.4.10 | Faculty | Director | Director | **PASS** | - |
| 3.14.4.11 | Faculty | Administrator | Administrator | **PASS** | - |
| 3.14.4.12 | Faculty | Approved | Approved | **PASS** | - |
| 3.14.4.13 | HOD | Not applied | Not applied | **PASS** | - |
| 3.14.4.14 | HOD | Director | Director | **PASS** | - |
| 3.14.4.15 | HOD | Administrator | Administrator | **PASS** | - |
| 3.14.4.16 | HOD | Approved | Approved | **PASS** | - |
| 3.14.4.17 | Director | Not applied | Not applied | **PASS** | - |
| 3.14.4.18 | Director | Administrator | Administrator | **PASS** | - |
| 3.14.4.19 | Director | Approved | Approved | **PASS** | - |

**3.14.5** **Test Result**

The function get\_status produced the output as expected for all the test cases

**3.15 CHECK FINANCE STATUS**

**3.15.1 Test items**

The unit to be tested here is the function get\_status (which facilitates the process of getting a financial application status), for the Undergrads and Research scholars.

This function is placed in the class Leave and hence can be used to display the status for both Undergrads and research scholars.

**3.15.2 Features to be tested**

The only feature to be tested for this function is the status it returns

**3.15.3 Item pass/ fail criteria**

If the correct status is displayed when invoked for a particular object, the test is deemed to have been passed.

|  |  |
| --- | --- |
| **Function Name** | get\_status |
| **Class** | Finance |
| **Invoked for** | **obj (Finance)** – Finance object with details of financial income, amount received currently, financial application status, etc |
| **Input Parameters** | None |
| **Processing** | None (since it is defined to be just a getter function) |
| **Output Parameters** | Return the string stored in the status field of the Finance object |
| **Code Snippet** | |
| **C:\Users\DELL\Pictures\Screenshots\Screenshot (136).png** | |

**3.14.4 Test cases**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test ID** | **Invoked for** | **Actual Application**  **Status** | **Actual Output** | **Result**  **(Pass/Fail)** | **Severity of the failure (if any)** |
| 3.14.4.1 | Undergrads | Not applied | Not applied | **PASS** | - |
| 3.14.4.2 | Undergrads | Administrator | Administrator | **PASS** | - |
| 3.14.4.3 | Undergrads | Approved | Approved | **PASS** | - |
| 3.14.4.4 | Research\_scholar | Not applied | Not applied | **PASS** | - |
| 3.14.4.5 | Research\_scholar | Faculty | Faculty | **PASS** | - |
| 3.14.4.6 | Research\_scholar | Administrator | Administrator | **PASS** | - |
| 3.14.4.7 | Research\_scholar | Approved | Approved | **PASS** | - |

**3.14.5** **Test Result**

The function get\_status produced the output as expected for all the test cases

**3.15 REGISTER**

**3.15.1 Test items**

The unit to be tested here is the function Register (which facilitates the process of registration).

* + 1. **Features to be tested**
* Uniqueness of ID
* Hierarchical registration

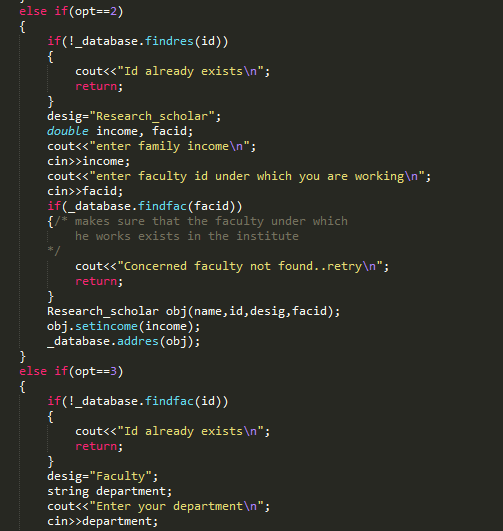
**3.15.3 Item pass/ fail criteria**

* The system should not allow more than one user of a particular designation to have the same ID
* Registration should be allowed only if the higher authorities exist in the database

|  |  |
| --- | --- |
| **Function Name** | Register |
| **Class** | Utility\_class |
| **Input Parameters**  **to the function** | None |
| **User inputs** | Choice of designation (from a menu) |
| Name |
| ID |
| **Processing** | * Checks if ID is already in use by another user of the same designation. * If ID is in use, registration with the inputted ID is not permitted. * Further, the system allows registration only if the higher authorities exist in the database.   Eg: A faculty of a department (say CSE) can register himself only if the CSE HOD has registered himself in the database.   * For an Undergrad user, family income is accepted during registration * For an Research\_scholar, family income and research supervisor’s ID are accepted during registration |
| **Output Parameters** | None |

**Code Snippet**

****

****



**3.15.4 Test cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **User** | **Inputted ID** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** | **Comments** |
| **-** | **Administrator** | **1234 (default)** | **-** | **-** | **-** | **-** |
| **-** | **Director** | **123**  **(default)** | **-** | **-** | **-** | **-** |
| 3.15.4.1 | Undergrads | 1 | Registration  Successful |  | **PASS** | - |
| 3.15.4.2 | Research\_  scholar | 2  (faculty ID  : 5) | Registration  Failed |  | **PASS** | Faculty not found |
| 3.15.4.3 | Faculty | 3  (Department : CSE) | Registration  Failed |  | **PASS** | HOD not found |
| 3.15.4.4 | HOD | 4  (Department : CSE) | Registration  Successful |  | **PASS** | - |
| 3.15.4.5 | Faculty | 3  (Department : EE) | Registration  Failed |  | **PASS** | HOD not found |
| 3.15.4.6 | Faculty | 3  (Department : CSE) | Registration  Successful |  | **PASS** | - |
| 3.15.4.7 | HOD | 4  (Department : EE) | Registration  Failed |  | **PASS** | User ID in use |
| 3.15.4.8 | HOD | 5  (Department : EE) | Registration  Successful |  | **PASS** | - |
| 3.15.4.9 | Faculty | 6  (Department : EE) | Registration  Successful |  | **PASS** | - |
| 3.15.4.10 | Research\_  scholar | 2  (faculty ID  : 3) | Registration  Successful |  | **PASS** | - |
| 3.15.4.11 | Research\_  scholar | 7  (faculty ID  : 6) | Registration  Successful |  | **PASS** | - |

**3.15.5** **Test Result**

The function get\_status produced the output as expected for all the test cases.

The 2 criteria:

* Uniqueness of ID and
* Hierarchical Registration

are satisfied

* 1. **LOGIN**

|  |  |
| --- | --- |
| **Function Name** | Login |
| **Class** | Utility\_class |
| **Input Parameters**  **to the function** | None |
| **User inputs** | Choice of designation (from a menu) |
| ID |
| **Processing** | * Checks if ID is already registered * If ID is registered, the control is handed over to the interface for the corresponding user. |
| **Output Parameters** | None |

**3.16.1 Test items**

The unit to be tested here is the function Login (which facilitates the process of logging in).

* + 1. **Features to be tested**

The system should allow a user to login, only if he has registered beforehand.

**3.16.3 Item pass/ fail criteria**

If the system allows a user to login, only if he has registered and generates an error message when the user is not registered, the test is deemed to have been passed.

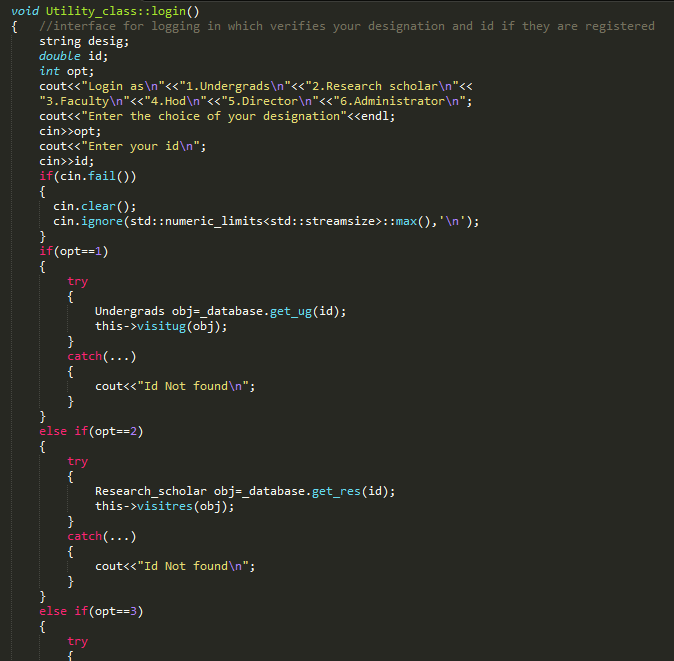
|  |  |
| --- | --- |
| **Present Situation of the Database** | |
| **User ID** | **Designation** |
| 123 | Director |
| 1234 | Administrator |
| 1 | HOD (CSE) |
| 2 | HOD (EE) |
| 3 | Faculty (CSE) |
| 4 | Faculty (EE) |
| 5 | Research\_scholar (Faculty: 3) |
| 6 | Research\_scholar (Faculty: 4) |
| 7 | Undergrads |

**3.16.4 Test Cases**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test ID** | **Inputted Designation** | **Inputted ID** | **Expected Result** | **Actual Result** | **Result**  **(Pass/Fail)** | **Comments** |
| 3.16.4.1 | Administrator | 1234 | Login Successful | Login Successful | **PASS** | **-** |
| 3.16.4.2 | Administrator | 1 | Login Failed | Login Failed | **PASS** | Incorrect ID |
| 3.16.4.3 | Director | 123 | Login Successful | Login Successful | **PASS** | **-** |
| 3.16.4.4 | Director | 1 | Login Failed | Login Failed | **PASS** | Incorrect ID |
| 3.16.4.5 | HOD | 1 | Login Successful | Login Successful | **PASS** | - |
| 3.16.4.6 | HOD | 2 | Login Successful | Login Successful | **PASS** | - |
| 3.16.4.7 | HOD | 3 | Login Failed | Login Failed | **PASS** | Incorrect ID |
| 3.16.4.8 | Faculty | 3 | Login Successful | Login Successful | **PASS** | - |
| 3.16.4.9 | Faculty | 4 | Login Successful | Login Successful | **PASS** | - |
| 3.16.4.10 | Faculty | 5 | Login Failed | Login Failed | **PASS** | Incorrect ID |
| 3.16.4.11 | Research\_  scholar | 5 | Login Successful | Login Successful | **PASS** | - |
| 3.16.4.12 | Research\_  scholar | 6 | Login Successful | Login Successful | **PASS** | - |
| 3.16.4.13 | Research\_  scholar | 7 | Login Failed | Login Failed | **PASS** | Incorrect ID |
| 3.16.4.14 | Undergrads | 7 | Login Successful | Login Successful | **PASS** | - |
| 3.16.4.15 | Undergrads | 8 | Login Failed | Login Failed | **PASS** | Incorrect ID |

**3.16.5 Test Result**

The function Login produced results as expected.

** Code Snippet**

****

**3.17**

1. **DETAILS OF SYSTEM TESTING**

This is the Testing performed on the Application under test

* To make sure that Critical Business scenarios were tested.
* To verify the entire application works as per the requirements.
* To verify that functionalities in the application work as intended without any errors.
* **Entry Criteria:**

1. The unit testing is performed for each module
2. For each user we take, we consider the case that he has already registered and hence logged in.

* **Exit criteria:**

1. All test cases are executed.
2. All defects if existent are removed.
3. The functionalities expected from the software are executing without failure.

## leave processing:

## Undergrad TO administrator (END-to-end)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Inputs** | **Intended Result** | **Actual Result** | **Pass/Fail** | **Comments** |
| 1 | UG: apply leave  **Example test case:**  Undergrads(mode):  Id:2  Days entered:20 | 1. Days: number of leaves applied 2. Leave: leave object associated with Undergrad | 1. Status: status of undergrad changes from “Not applied” to “Administrator” if Days<max\_leaves\_premitted 2. The leave object of the applicant is inserted in the list of leaves pending which needs to be approved by administrator | 1. Same as expected result 2. Same as expected result | Pass  Pass | The leave application applied by undergrad is processed to Administrator if it satisfies the conditions as mentioned in unit testing. |
| 2 | Administrator: approve leave  **Example test case:**  Administrator(mode)  For approval:  Selected id:2  Selected designation: Undergrads | Id of leave applicant  Designation of leave applicant | 1. Status: status of undergrad changes from “Administrator” to “Approved” 2. The leave application object is removed from leave pending list of administrator 3. Max leaves: of the applicant now reduces from initial number by the amount of leaves which he has applied and are now approved | 1. Same as expected 2. Same as expected 3. Same as expected | Pass  Pass  3.Pass | Once the administrator approves the concerned leave application request of applicant, leave is approved and maximum leaves remaining are deducted |

## leave processing:

## Research scholar TO administrator (END-tO-End and Pairwise)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Inputs** | **Intended Result** | **Actual Result** | **Pass/Fail** | **Comments** |
| 1 | Research\_scholar: apply leave  **Example test case:**  Reserach\_scholar (mode):  Id:12  Days entered:10 | 1. Days: number of leaves applied 2. Leave: leave object associated with research\_scholar | 1. Status: status of research scholar changes from “Not applied” to “Faculty” if Days < max\_leaves\_premitted 2. The leave object of the applicant is inserted in the list of leaves pending which needs to be approved by faculty advisor of the research scholar | 1. Same as expected result 2. Same as expected result | Pass  Pass | The leave application of a research scholar is forwarded to his faculty advisor if it satisfies the conditions as mentioned in unit testing. |
| 2 | Faculty: approve leave  (Faculty who is the advisor of concerned research scholar)  **Example test case:**  Faculty(mode)  For approval:  Selected id:12  Selected designation: Research\_scholar | 1. Id of leave applicant 2. Designation of applicant | 1. Status: status of research scholar changes from “Faculty” to “Administrator” 2. The leave application object is removed from leave pending list of faculty advisor and inserted in the list of leaves pending which needs to be approved by administrator | 1. Same as expected 2. Same as expected | Pass  Pass | Once the faculty advisor approves the application of applicant, leave is processed and forwarded to list of pending leaves of administrator |
| 3 | Administrator: approve leave  **Example test case:**  Administrator(mode)  For approval:  Selected id:12  Selected designation: Research\_scholar | 1. Id of leave applicant 2. Designation of leave applicant | 1. Status: status of scholar changes from “Administrator” to “Approved” 2. The leave application object is removed from leave pending list of administrator 3. Max leaves: of the applicant now reduces from initial number by the amount of leaves which he has applied and are now approved | 1. Same as expected 2. Same as expected 3. Same as expected | Pass  Pass  3.Pass | Once the administrator approves the concerned leave application request of applicant, leave is approved and maximum leaves remaining are deducted |

## leave processing:

## FAculty TO administrator (END-to-end and pairwise)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Inputs** | **Intended Result** | **Actual Result** | **Pass/Fail** | **Comments** |
| 1 | Faculty: apply leave  **Example test case:**  Faculty(mode):  Id:1  Days entered:15 | 1. Days: number of leaves applied 2. Leave: leave object associated with faculty | 1. Status: status of faculty changes from “Not applied” to “ HOD ” if Days<max\_leaves\_premitted 2. The leave object of the applicant is inserted in the list of leaves pending which needs to be approved by HOD of the faculty | 1. Same as expected result  2.Same as expected result | Pass  Pass | The leave application applied by faculty is processed to his HOD if it satisfies the conditions as mentioned in unit testing. |
| 2 | HOD: approve leave  ( HOD of department in which faculty works)  **Example test case:**  HOD (mode)  For approval:  Selected id:1  Selected designation: Faculty | 1. Id of leave applicant 2. Designation of   applicant | 1. Status: status of faculty changes from “ HOD ” to “Administrator” 2. The leave application object is removed from the list of pending leaves of the HOD and inserted in the list of pending leaves which needs to be approved by administrator | 1. Same as expected  2. Same as expected | Pass  Pass | Once the HOD approves the concerned leave application request of applicant, leave is processed forward to list of pending leaves of administrator |
| 3 | Administrator: approve leave  **Example test case:**  Administrator(mode)  For approval:  Selected id:1  Selected designation: Faculty | 1. Id of leave applicant 2. Designation of leave applicant | 1. Status: status of faculty changes from “Administrator” to “Approved” 2. The leave application object is removed from leave pending list of administrator 3. Max leaves: of the applicant now reduces from initial number by the amount of leaves which he has applied and are now approved | 1. Same as expected  2. Same as expected  3.Same as expected | Pass  Pass  3.Pass | Once the administrator approves the concerned leave application request of applicant, leave is approved and maximum leaves remaining are deducted |

## leave processing:

## Hod TO administrator (END-to-end and pairwise)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Inputs** | **Intended Result** | **Actual Result** | **Pass/Fail** | **Comments** |
| 1 | HOD: apply leave  **Example test case:**  HOD (mode):  Id:5  Days entered:5 | 1. Days :number of leaves applied 2. Leave: leave object associated with HOD | 1. Status: status of undergrad changes from “Not applied” to “Director” if Days < max\_leaves\_premitted 2. The leave object of the applicant is inserted in the list of leaves pending which needs to be approved by the Director | 1. Same as expected result  2.Same as expected result | Pass  Pass | The leave application applied by HOD is processed to director if it satisfies the conditions as mentioned in unit testing. |
| 2 | Director: approve leave  **Example test case:**  Director (mode)  For approval:  Selected id:5  Selected designation: HOD | 1. Id of leave applicant 2. Designation of   applicant | 1. Status: status of HOD changes from “Director” to “Administrator” 2. The leave application object is removed from leave pending list of Director and inserted in the list of leaves pending which needs to be approved by administrator | 1. Same as expected  2. Same as expected | Pass  Pass | Once the director approves the concerned leave application request of applicant, leave is processed forward to list of pending leaves of administrator |
| 3 | Administrator: approve leave  **Example test case:**  Administrator(mode)  For approval:  Selected id:12  Selected designation: HOD | 1. Id of leave applicant 2. Designation of leave applicant | 1. Status: status of scholar changes from “Administrator” to “Approved” 2. The leave application object is removed from leave pending list of administrator 3. Max leaves: of the applicant now reduces from initial number by the amount of leaves which he has applied and are now approved | 1. Same as expected  2. Same as expected  3.Same as expected | Pass  Pass  3.Pass | Once the administrator approves the concerned leave application request of applicant, leave is approved and maximum leaves remaining are deducted |

## leave processing:

## Director TO administrator(END-to-end)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Inputs** | **Intended Result** | **Actual Result** | **Pass/Fail** | **Comments** |
| 1 | Director: apply leave  **Example test case:**  Director(mode):  Id:123(only possible id)  Days entered:22 | 1. Days: number of leaves applied 2. Leave: leave object associated with Director | 1. Status:status of director changes from “Not applied” to “Administrator” if Days < max\_leaves\_premitted 2. The leave object of the applicant is inserted in the list of leaves pending which needs to be approved by administrator | 1. Same as expected result  2.Same as expected result | Pass  Pass | The leave application applied by director is processed to Administrator if it satisfies the conditions as mentioned in unit testing. |
| 2 | Administrator: approve leave  **Example test case:**  Administrator(mode)  For approval:  Selected id:123  Selected designation: Director | 1. Id of leave applicant 2. Designation of leave applicant | 1. Status: status of director changes from “Administrator” to “Approved” 2. The leave application object is removed from leave pending list of administrator 3. Max leaves: of the applicant now reduces from initial number by the amount of leaves which he has applied and are now approved | 1. Same as expected  2. Same as expected  3.Same as expected | Pass  Pass  3.Pass | Once the administrator approves the concerned leave application request of applicant, leave is approved and maximum leaves remaining are deducted |

## Finance processing:

## Undergrad TO administrator (END-to-end)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Inputs** | **Intended Result** | **Actual Result** | **Pass/Fail** | **Comments** |
| 1 | UG: apply finance  **Example test case:**  Undergrads(mode):  Id:2  amount entered:2000 | 1. Amount: total financial assistance sought 2. Finance: finance object associated with Undergrad | * 1. Status: status of undergrad changes from “Not applied” to “Administrator” if amount applied + current scholarship < upper bound and family income is within given bracket   2. The finance object of the applicant is inserted in the list of finances pending which needs to be approved by administrator | 1. Same as expected result 2. Same as expected result | Pass  Pass | The finance application applied by undergrad is processed to Administrator if it satisfies the conditions as mentioned in unit testing. |
| 2 | Administrator: approve finance  **Example test case:**  Administrator(mode)  For approval:  Selected id:2  Selected designation: Undergrads | 1. ID of finance applicant 2. Designation of finance applicant | 1. Status: status of undergrad changes from “Administrator” to “Approved” 2. The finance application object is removed from the list of pending finances of administrator 3. Amount withdrawn:   The amount received as financial assistance is updated by adding to it, the amount which has been approved now. | 1. Same as expected 2. Same as expected 3. Same as expected | Pass  Pass  3.Pass | Once the administrator approves the concerned finance application request of applicant, finance is approved and amount withdrawn is updated accordingly |

## Finance processing:

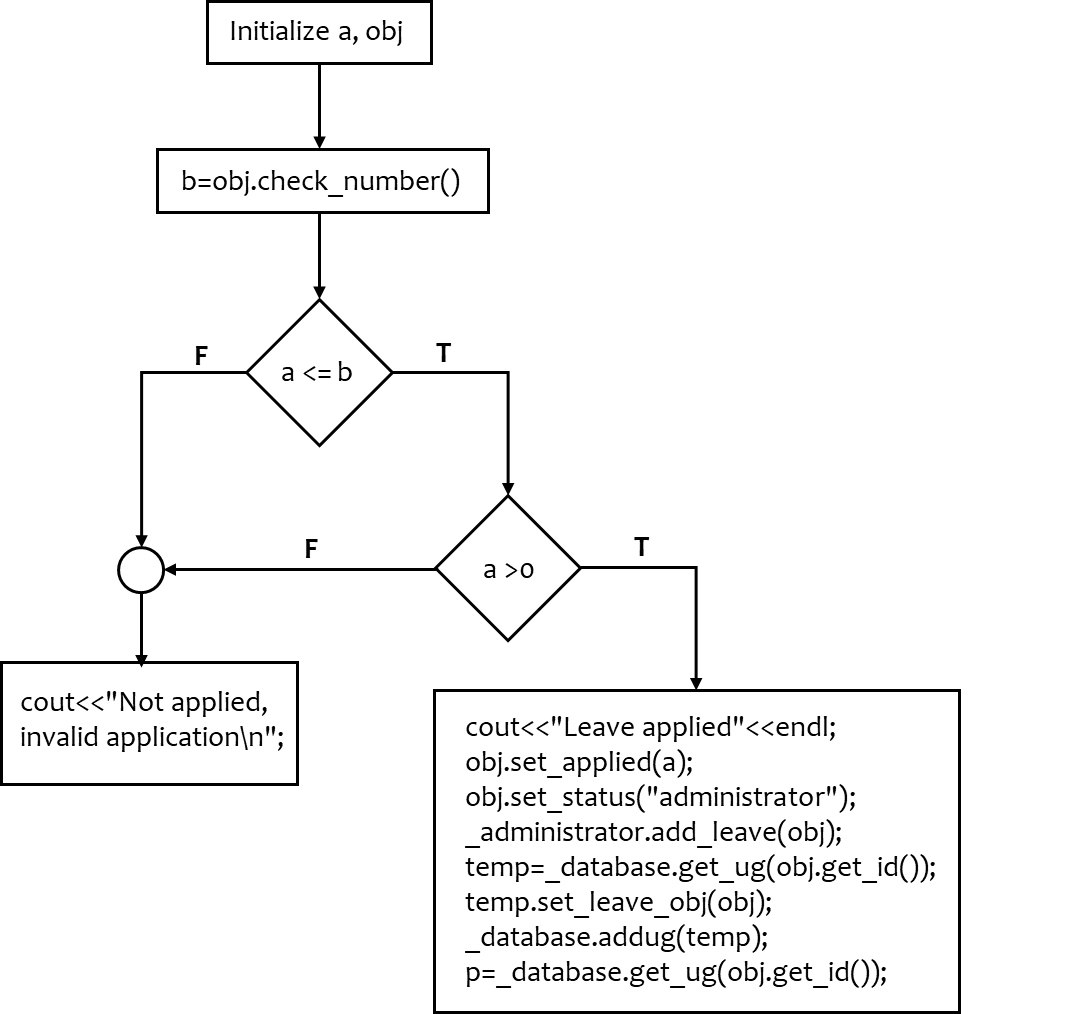
## Research scholar TO administrator (END-to-end and Pairwise)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Step** | **Test Steps** | **Inputs** | **Intended Result** | **Actual Result** | **Pass/Fail** | **Comments** |
| 1 | Research\_scholar: apply finance  **Example test case:**  Research\_scholar(mode):  Id:5  amount entered:5000 | 1. Amount :total financial assistance sought 2. Finance: finance object associated with Undergrad | 1. Status: status of scholar changes from “Not applied” to “faculty if amount applied + current scholarship<upper bound and family income is within given bracket 2. The finance object of the applicant is inserted in the list of pending finances which need to be approved by Faculty | 1. Same as expected result  2.Same as expected result | Pass  Pass | The finance application applied by research scholar is processed to his faculty advisor if it satisfies the conditions as mentioned in unit testing.. |
| 2 | Faculty: approve finance  (concerned faculty advisor of the research scholar)  **Example test case:**  Faculty (mode)  For approval:  Selected id:5  Selected designation: Research\_scholar | 1. ID of finance applicant 2. Designation of finance applicant | 1. Status: status of scholar changes from “faculty” to “Administrator”. 2. The finance application object is removed from list of pending finances of faculty and inserted into the list of pending finances of administrator | 1. Same as expected  2. Same as expected | Pass  Pass | Once the faculty approves the application request, finance is processed and forwarded to the list of pending finances of administrator |
| 3 | Administrator: approve finance  **Example test case:**  Administrator(mode)  For approval:  Selected id:12  Selected designation: Research\_scholar | 1. ID of finance applicant 2. Designation of finance applicant | 1. Status: status of scholar changes from “Administrator” to “Approved” 2. The finance application object is removed from the list of pending finances of administrator 3. Amount withdrawn:   The amount received as financial assistance is updated by adding to it, the amount which has been approved now. | 1. Same as expected  2. Same as expected  3.Same as expected | Pass  Pass  3.Pass | Once the administrator approves the application request, finance is approved and amount withdrawn is updated accordingly |

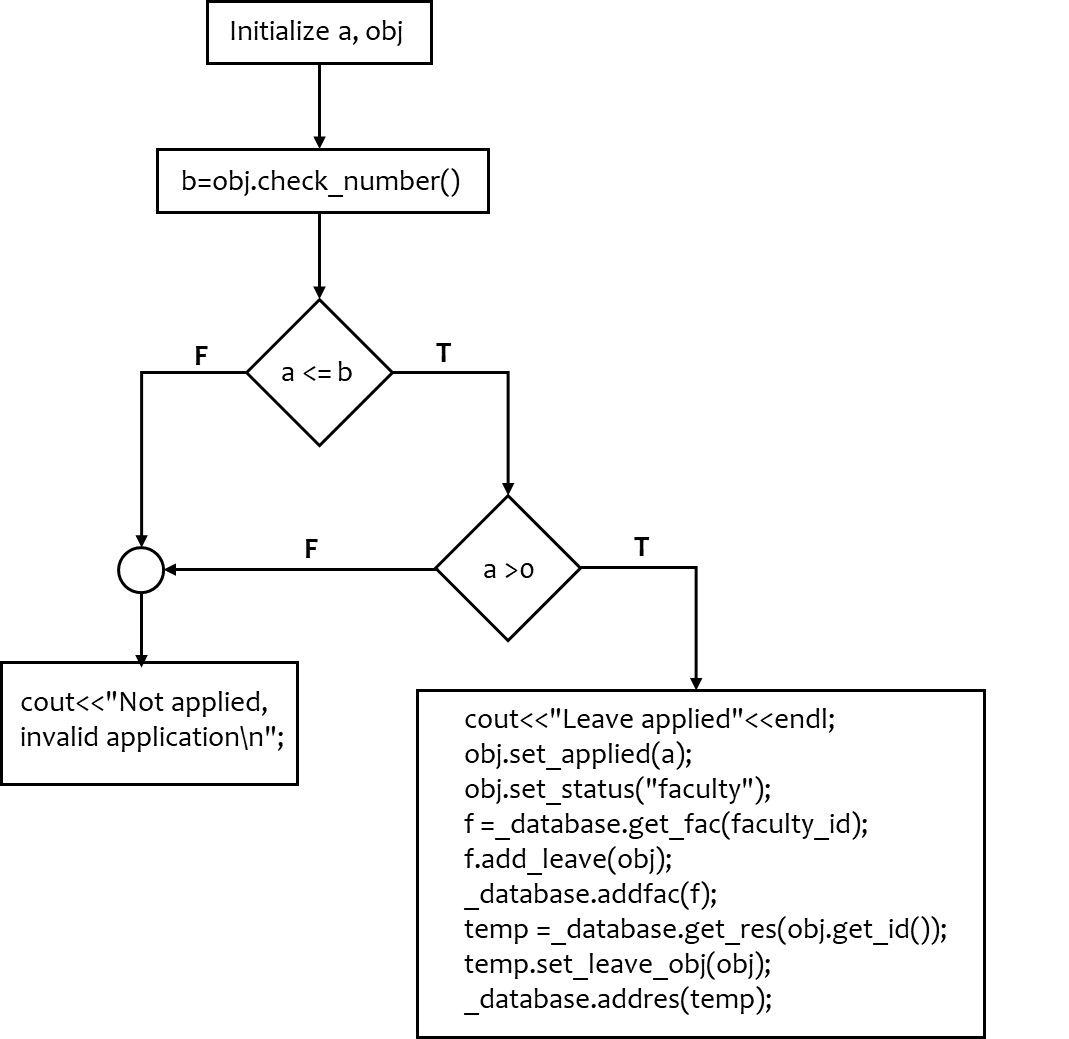
1. **APPENDIX**

***Control Flow Graphs***

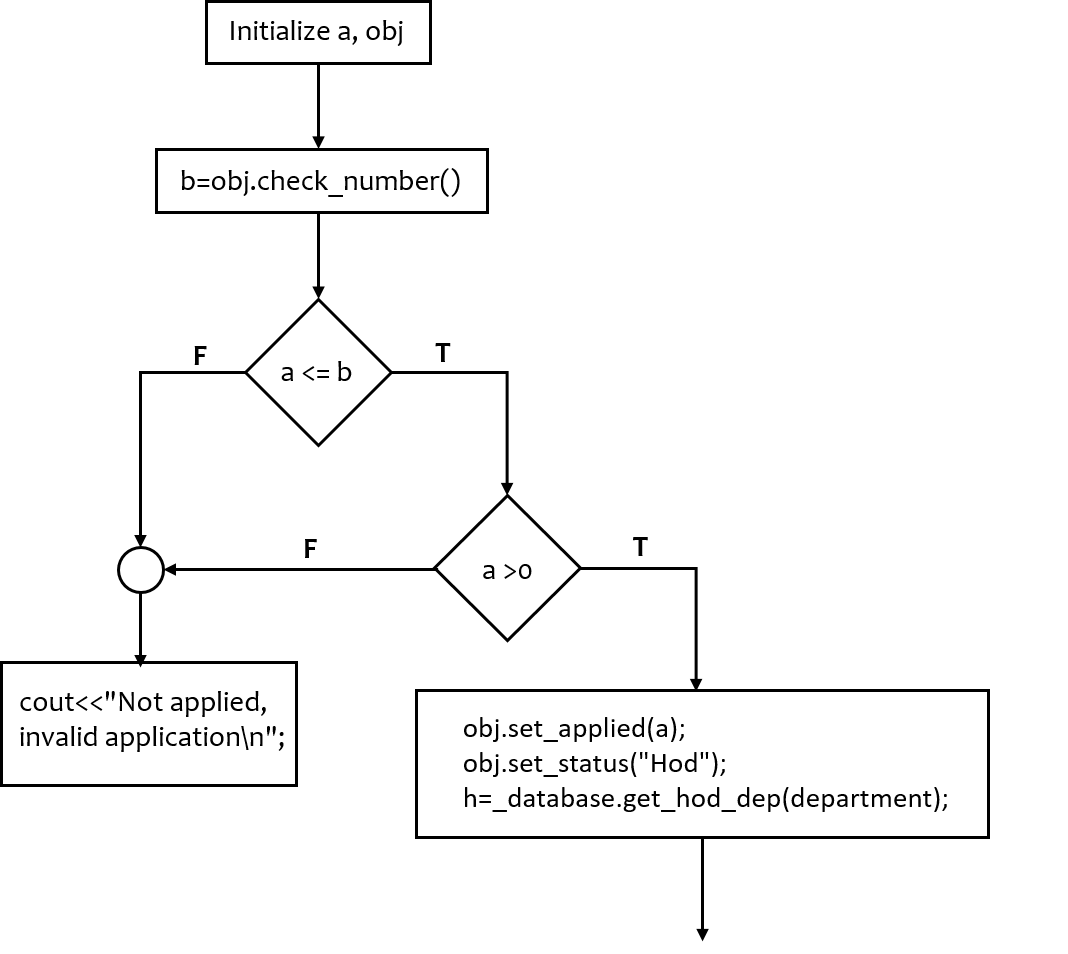
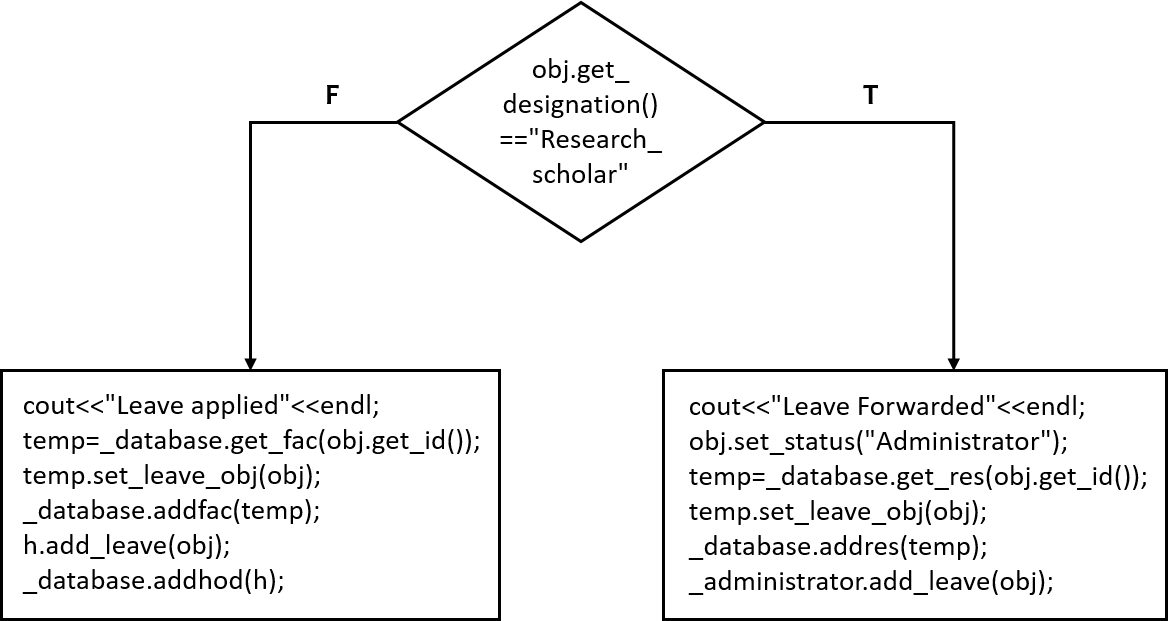
1. apply\_leave for Undergrads

****

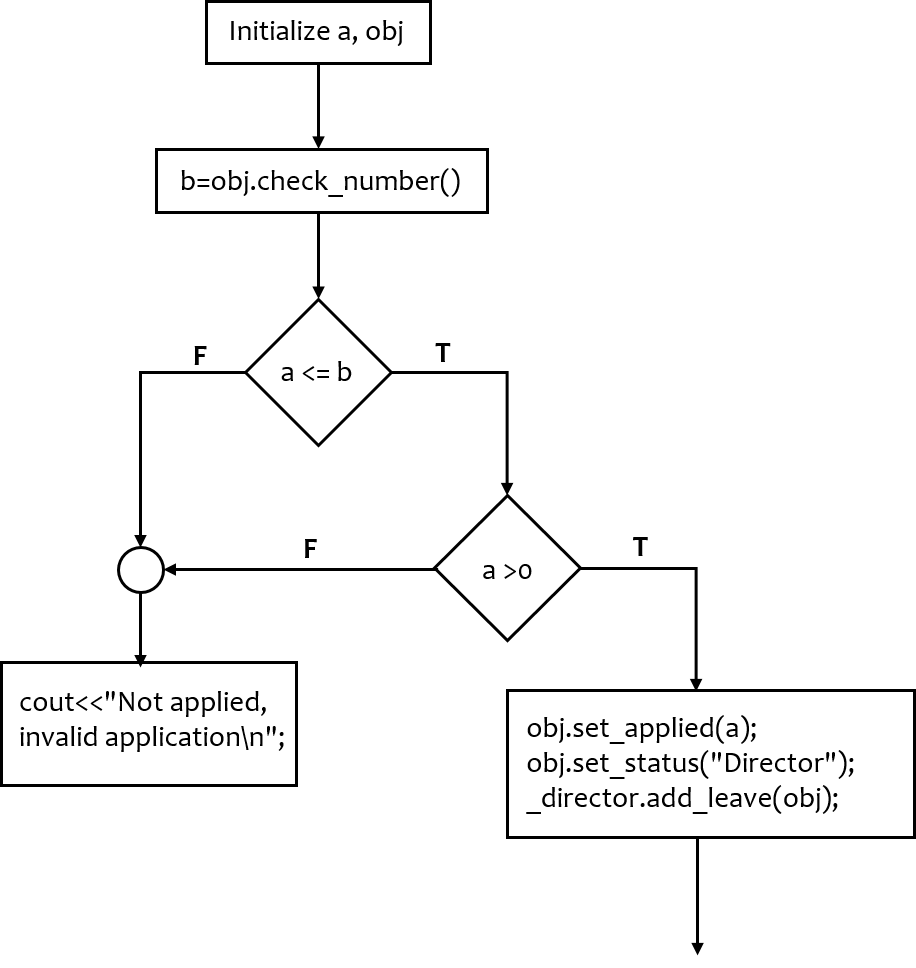
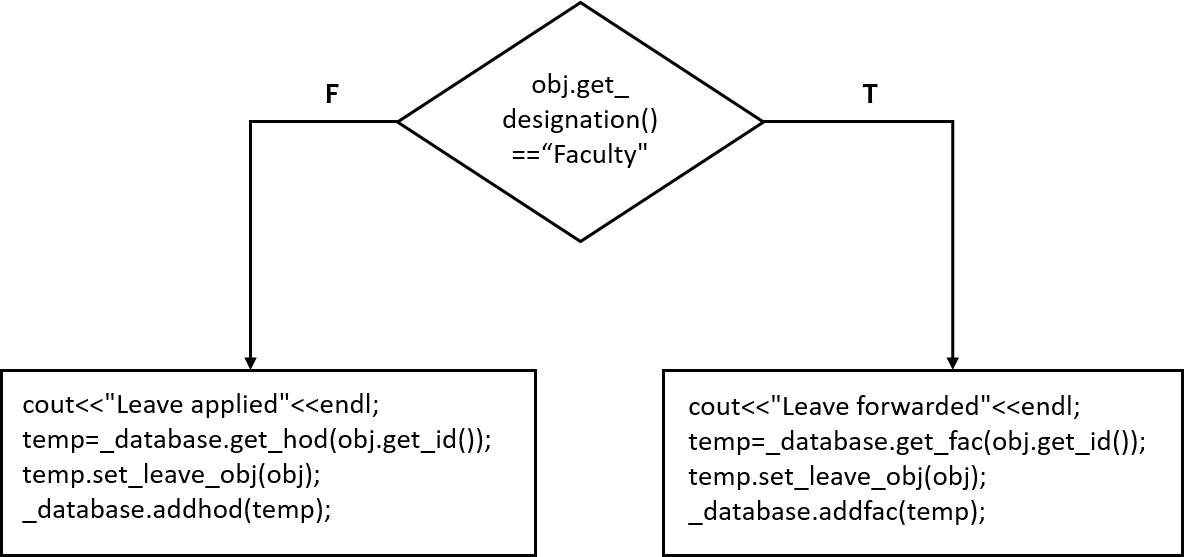
1. apply\_leave for Research\_scholar

****

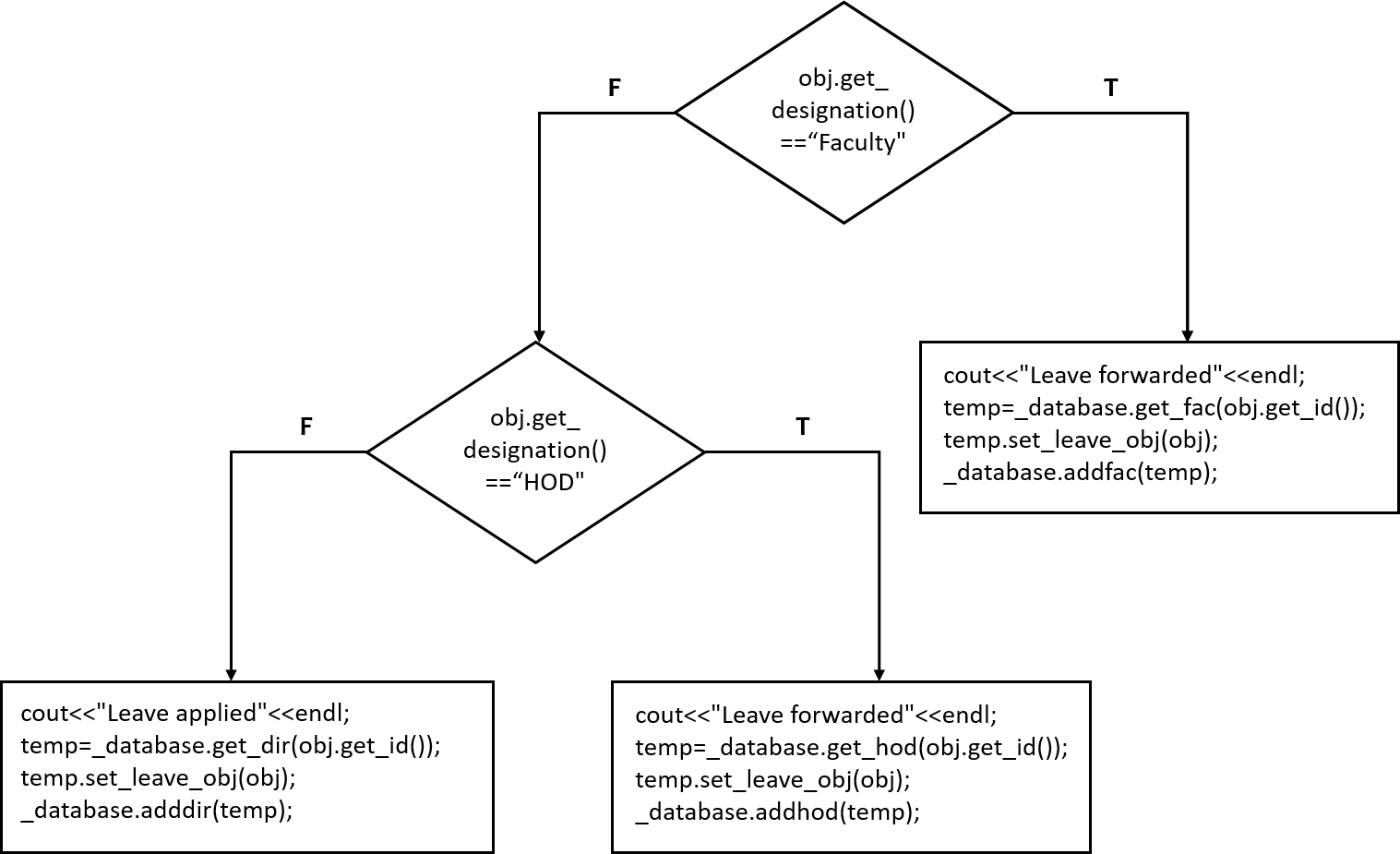
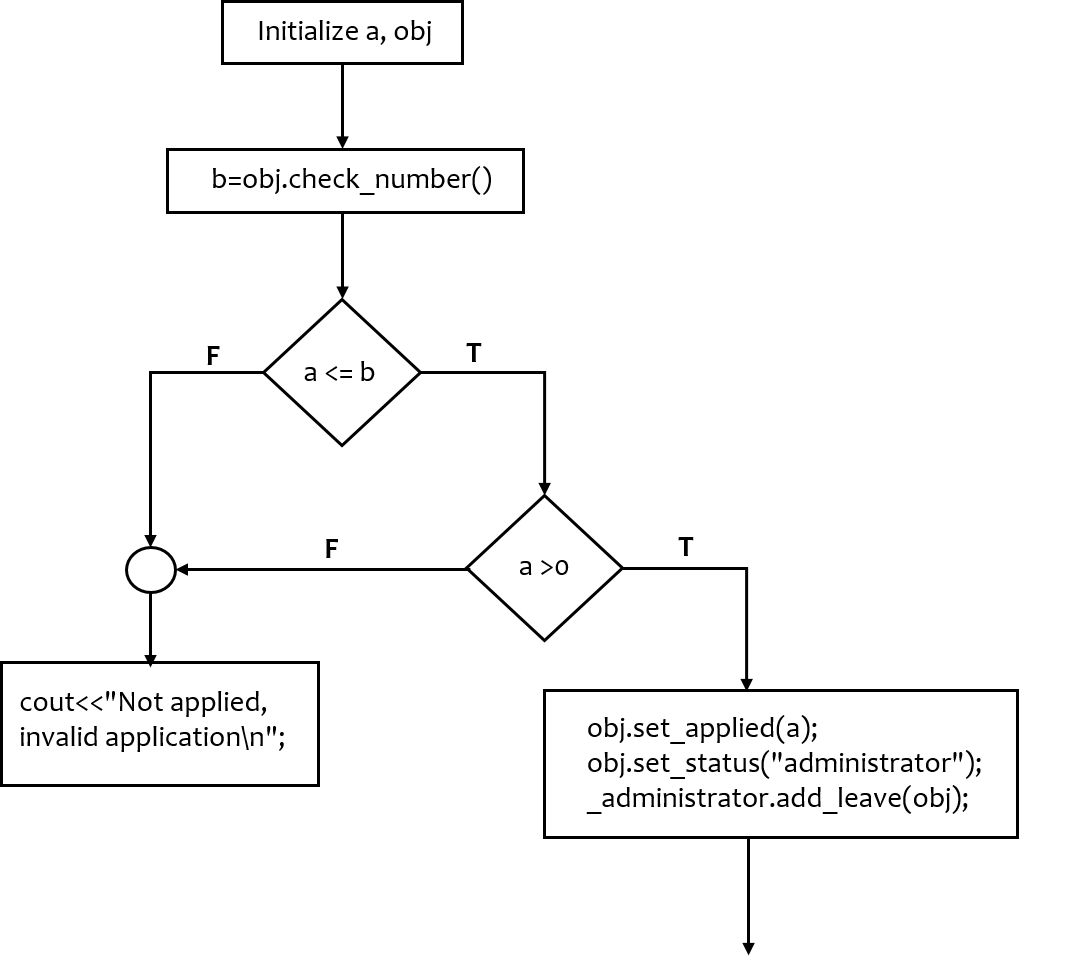
1. apply\_leave for Faculty



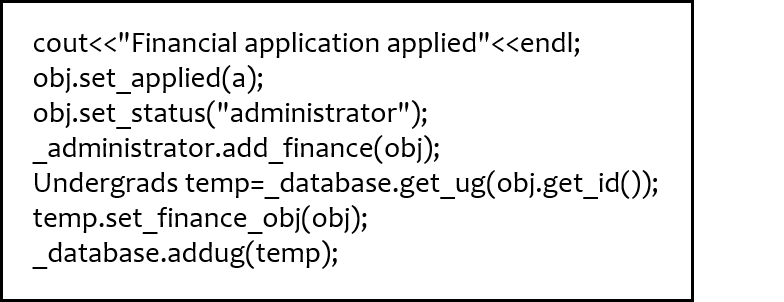
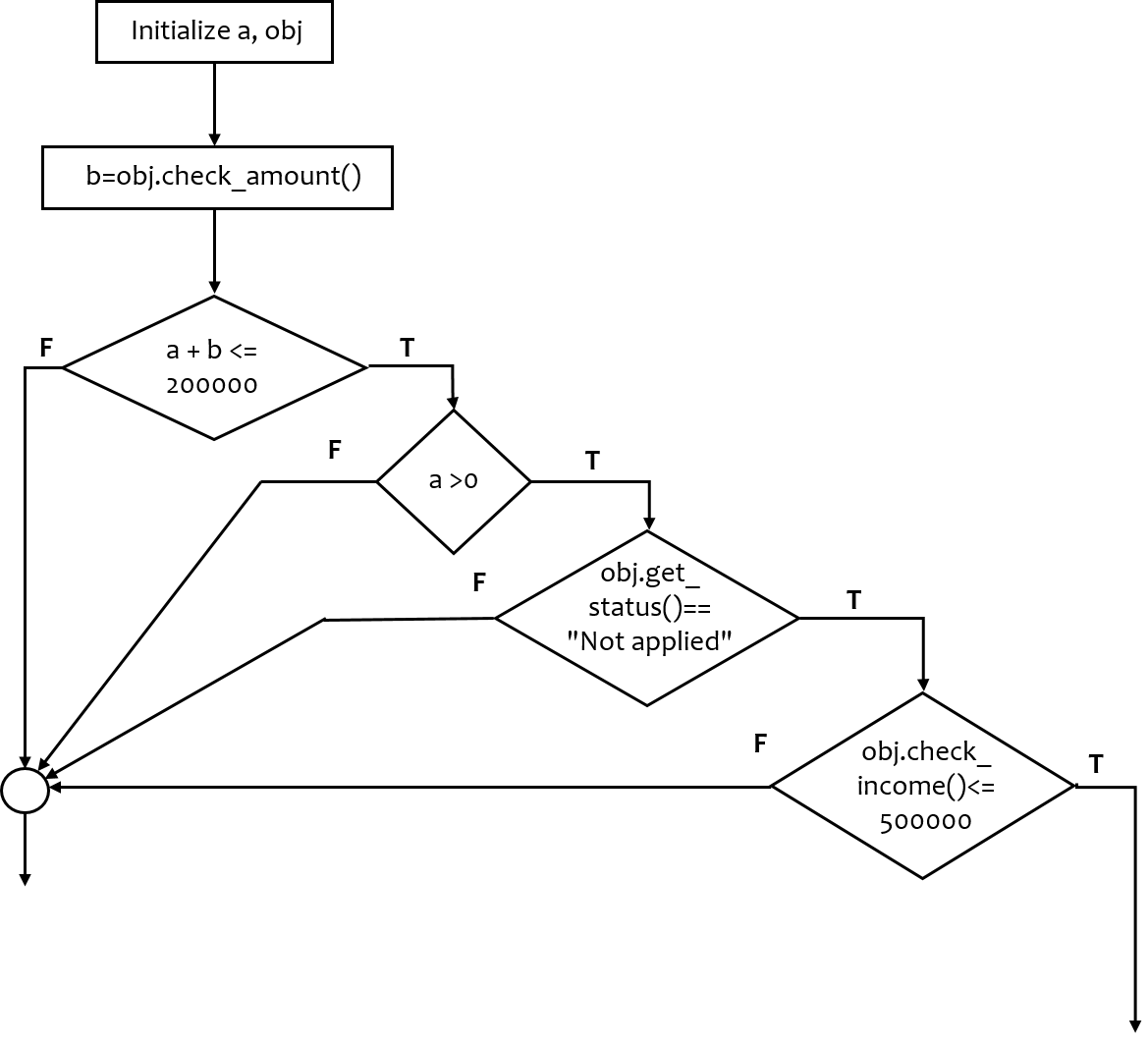
1. apply\_leave for HOD



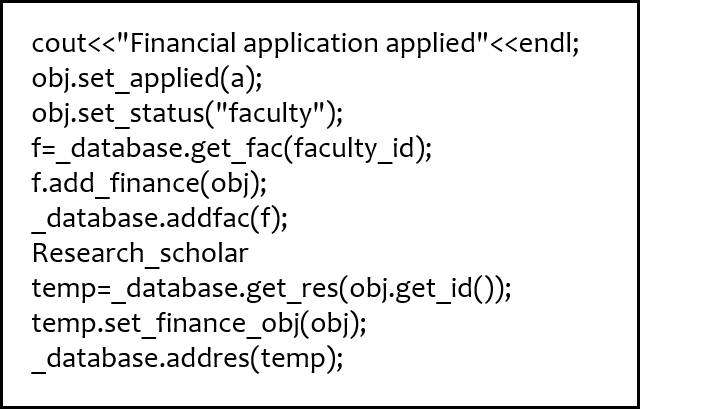
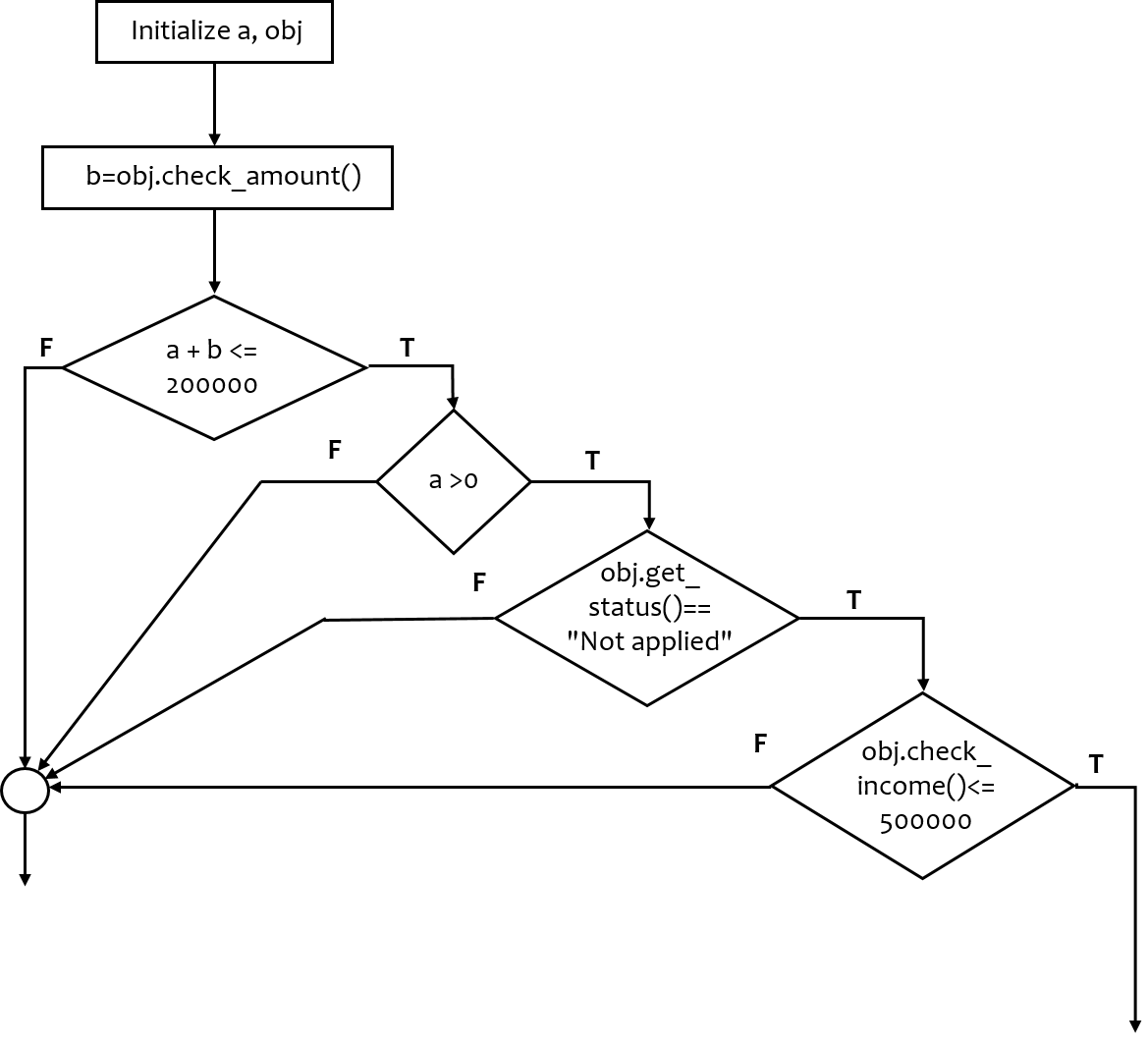
1. apply\_leave for HOD



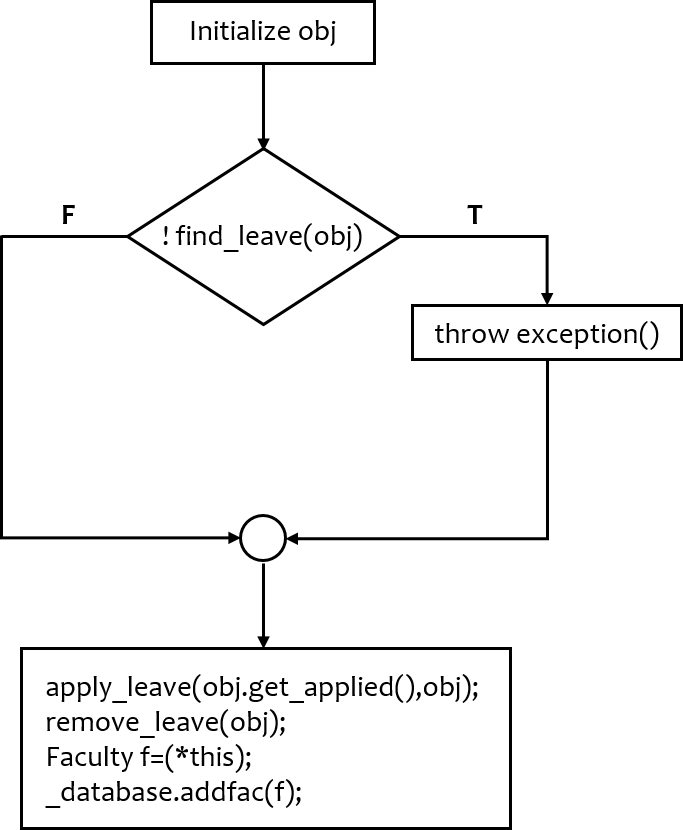
1. apply\_finance for Undergrads



1. apply\_finance for Research\_scholar

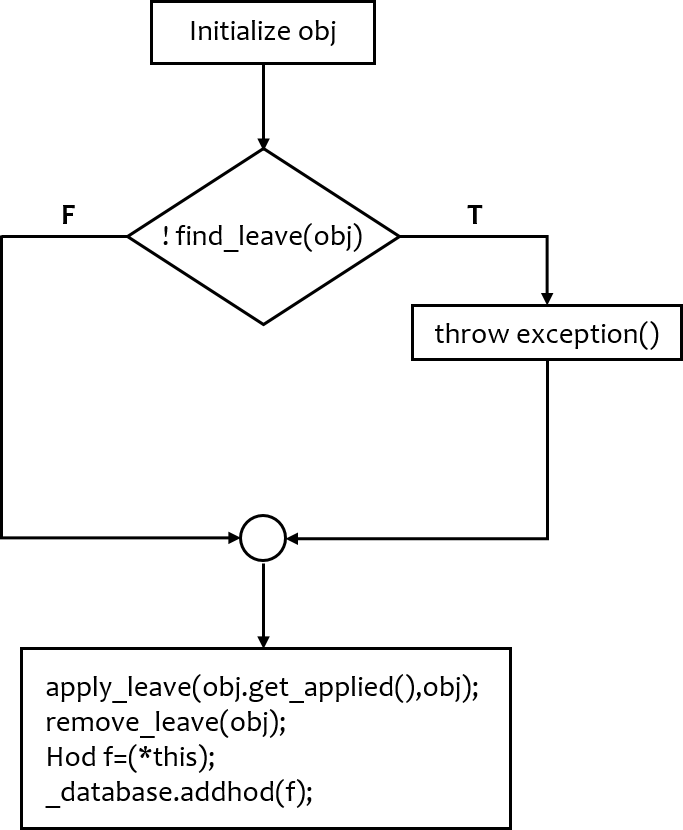


* 1. approve\_leave for Faculty



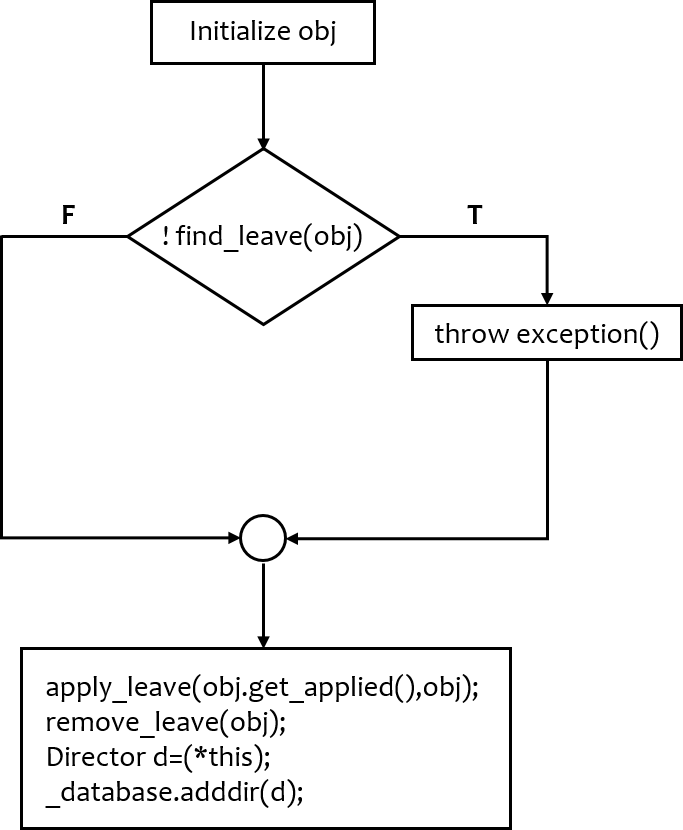
Control is handed over to the apply\_leave function for Faculty with number of leaves and obj as parameters

1. approve\_leave for HOD



Control is handed over to the apply\_leave function for HOD with number of leaves and obj as parameters

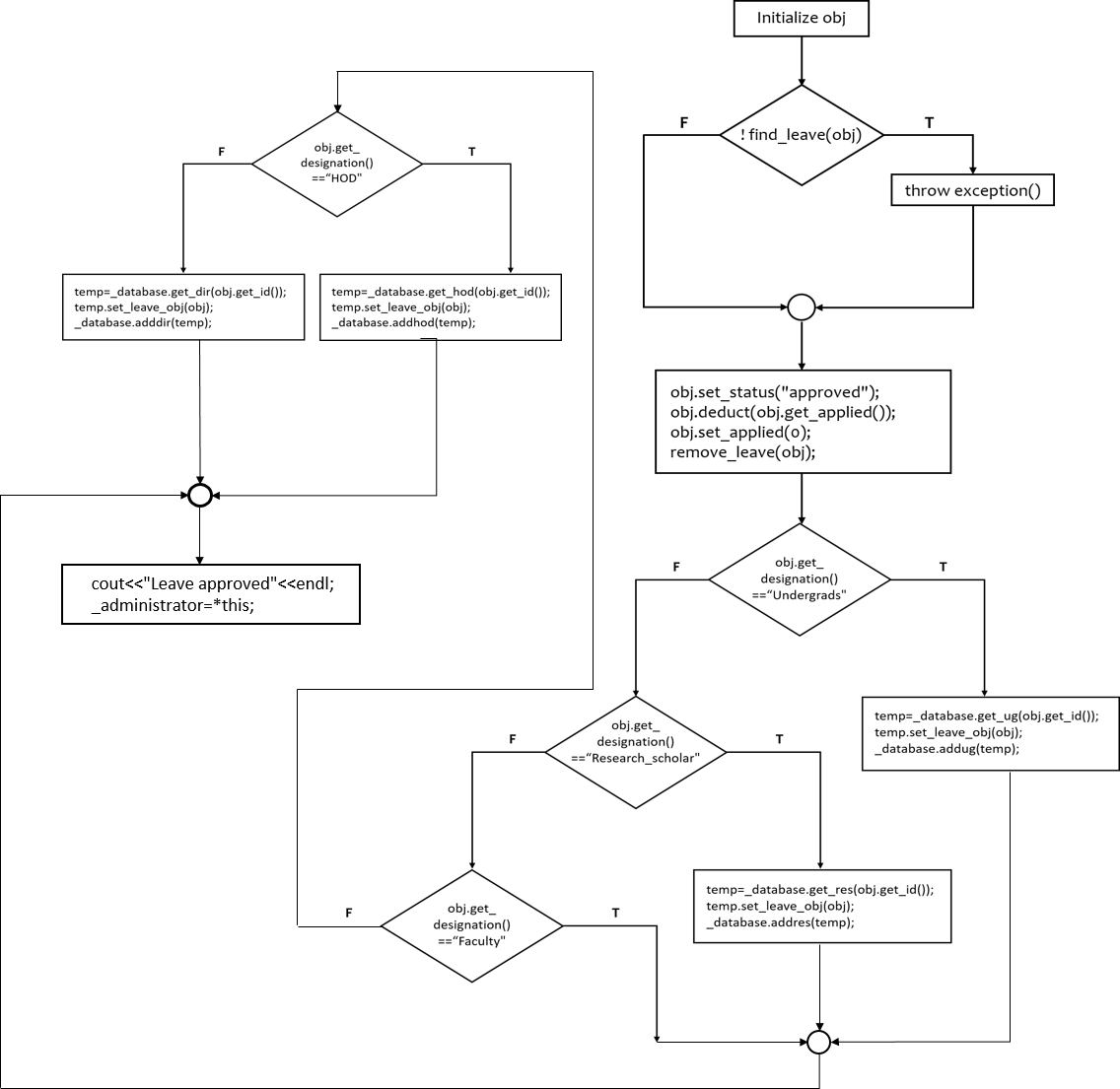
1. approve\_leave for Director



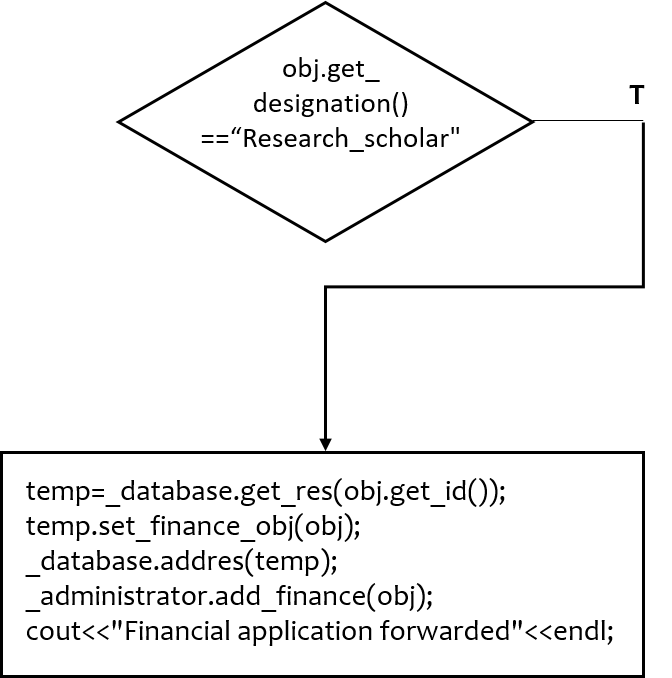
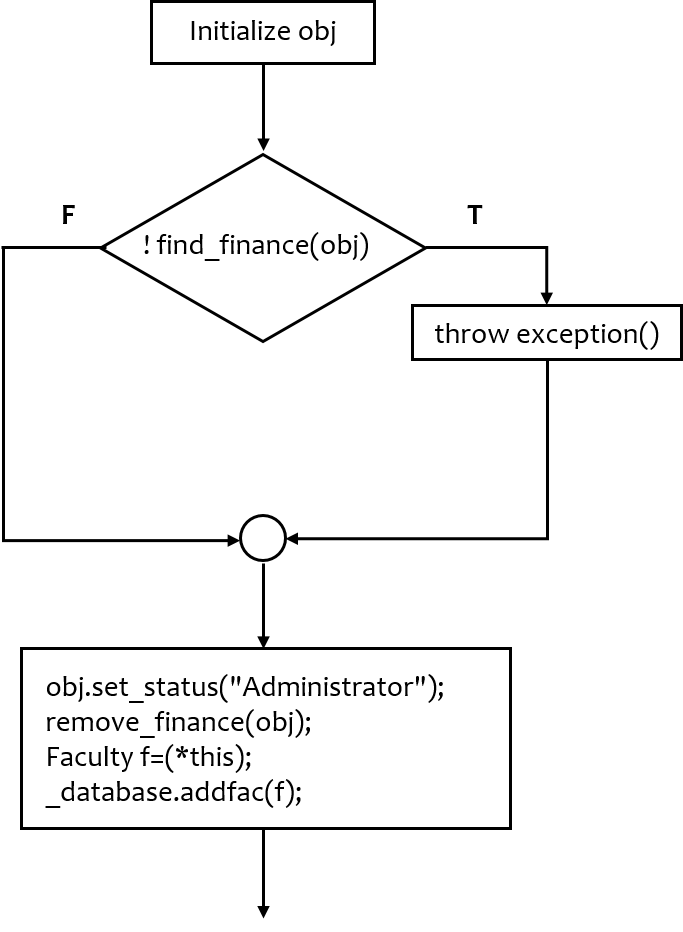
Control is handed over to the apply\_leave function for the Director with number of leaves

and obj as parameters

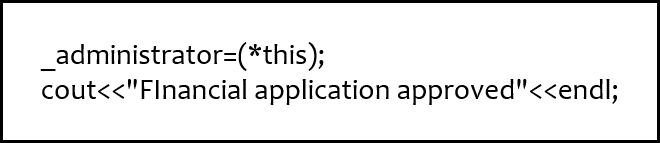
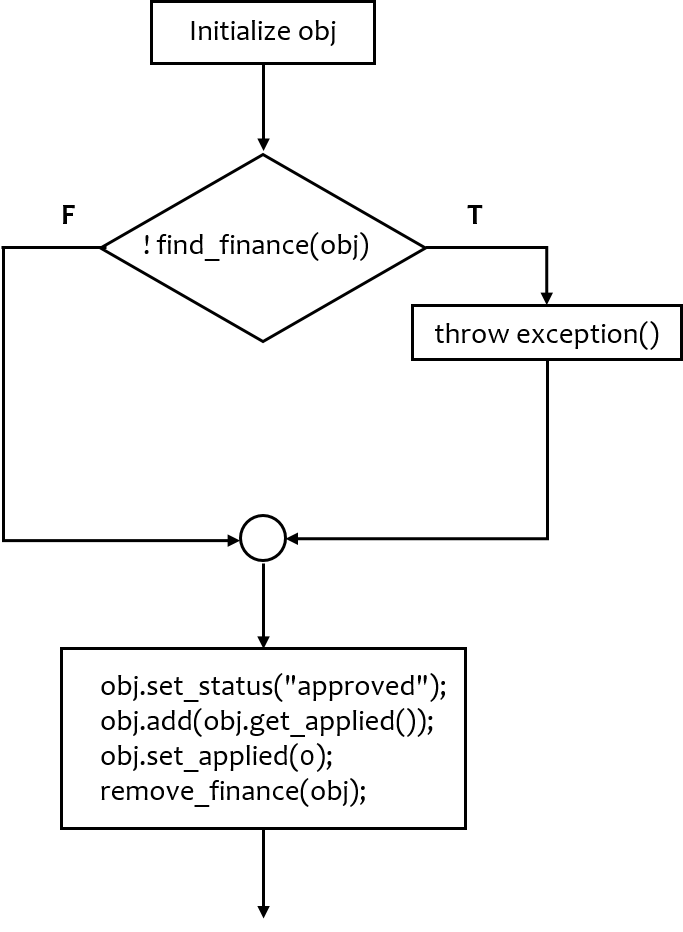
1. approve\_leave for Administrator



1. approve\_finance for Faculty



1. approve\_finance for Administrator



***Glimpses into the execution …***

